This item was submitted to Loughborough's Research Repository by the author.
Items in Figshare are protected by copyright, with all rights reserved, unless otherwise indicated.

Investigation on soft computing techniques for airport environment evaluation systems

PLEASE CITE THE PUBLISHED VERSION

PUBLISHER

© Yingjie Yang

PUBLISHER STATEMENT

This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) licence. Full details of this licence are available at: https://creativecommons.org/licenses/by-nc-nd/4.0/

LICENCE

CC BY-NC-ND 4.0

REPOSITORY RECORD

Yang, Yingjie. 2019. “Investigation on Soft Computing Techniques for Airport Environment Evaluation Systems”. figshare. https://hdl.handle.net/2134/35015.

Loughborough
University

University Library

Author/Filing Title YANG, Y

Class Mark T

Please note that fines are charged on ALL
overdue items.

040358650X

Investigation on Soft Computing Techniques for Airport Environment Evaluation Systems

By

Yingjie Yang

Submitted in partial fulfilment of the requirements

for the award of

Doctor of Philosophy at Loughborough University

Feb 2008

©by Yingjie Yang 2008

Loughborough University
Pilkington Library

Date 16/6/09

Class T

Acc No. 040352650X

Abstract

Spatial and temporal information exist widely in engineering fields, especially in airport environmental management systems. Airport environment is influenced by many different factors and uncertainty is a significant part of the system. Decision support considering this kind of spatial and temporal information and uncertainty is crucial for airport environment related engineering planning and operation. Geographical information systems and computer aided design are two powerful tools in supporting spatial and temporal information systems. However, the present geographical information systems and computer aided design software are still too general in considering the special features in airport environment, especially for uncertainty. In this thesis, a series of parameters and methods for neural network based knowledge discovery and training improvement are put forward, such as the relative strength of effect, dynamic state space search strategy and compound architecture. Fuzzy sets are adopted to support the sustainability evaluation, and a hierarchical pyramid architecture is proposed to simulate the decision process of human panel meetings. For uncertainty in geometrical object, a grey geometry representation using grey systems is developed. The union, intersection and difference operation are also defined. Based on these work, a neural network based noise, emission and waste model is established which is completely different from the present mainstream models. In addition to these works, the user friend interface for 3D airport operation planning is also explored and a convenient computer aided design facility is combined into the prototype. With a set of available noise data from Manchester airport, we verified some of our proposed models and the result is very promising.

Key words: Neural Networks, Intuitionistic Fuzzy Sets, Grey Geometry, Rough Sets, Airport Noise, Airport Environment Evaluation Systems

Contents

List of Figures

List of Tables

Acknowledgements

It was my good fortune to have Dr. Chris Hinde as my advisor while at Loughborough University. He taught me not only on how to work in my research but also a good attitude in my research career. Chris was always there when I needed him and it was his encouragement that kept me completing this research.

I would also like to thank Dr. David Gillingwater for his great support to my research and his valuable advices in my work. David's support is also an essential part of my success in this research.

I greatly appreciate the help of my friends at De Montfort University and Manchester Metropolitan University. Professor Robert John and Dr. Francisco Chiclana gave me support in my work on uncertainty models. Professor C. Thomas, Dr. D. Raper, and Dr. P. Upham and our partners at Manchester airport provided data for the research.

Thanks go to my family, my wife Yanling Wang and my daughter Jing Yang. They gave me consistent support to my work and that was essential for me to keep full energy in this research.

Chapter 1

Introduction

1.1 Challenge to airport environment models

As transportation hubs, airports comprise many modes of air and surface transport. Changes in the magnitude of these operations tend directly and indirectly to lead to corresponding social and environmental change, not least of which are employment opportunities and adverse environmental impacts [133]. On the one hand, society needs economic development to meet its social objectives, and a healthy natural environment is a precondition for society's existence. The resulting tension gives rise to the necessity for sustainable airport development [78], rather than airport development at undue environmental cost. Sustainable airport development requires a carefully planned balance between socio-economic and environmental objectives. The scale and risks of airport operations mean that operational decisions require careful planning before practical implementation. Similarly, sustainable development concerns prevention of the potential degradation of the foundation of the existence for human society, and hence also needs to be taken into account before the construction of airport-related infrastructure. This planning has to be based on a realistic analysis of the operational and environmental data monitored from airport operations. In this sense, a comprehensive decision support system is a necessary part of sustainable airport development. The decision support systems have to be able to simulate different design and operation at airports. Different from other transportation modes, airport operation involves not only the 2D operation of surface transport, its 3rd dimension is significant for air side operation. Therefore, a fully functional Computer Aided Design (CAD) facility for the 3D operation at airport is necessary in addition to the 2D facility in traditional GIS-T[125]. In addition to this, a decision support system for airport environment has to accommodate those uncertainties existing in the operation of airports. With the rapid development of airport relevant models, many models have been developed for airport operation and environment, such as SIMMOD (local airspace, runway and apron planning) [23] INM [41] (noise exposure) and EDMS [33] (concentration and distribution of gaseous pollutants). However, each model focuses on some specific part of the system and provide separate and limited design facility. They represent the same operation in different formats and provide very limited data input facility. The input data in one model is usually not applicable to the other models. The learning curves for these models are not flat because of their limited input facility. More importantly, the significance of uncertainty has not gained enough attention in these systems at all. Most derivation of their data come from some mathematical models under some unrealistic assumption about airports. In fact, each airport operates on a unique geographical environment which is completely different from other airports. The weather condition can change significantly between airports and the local residence may have different reaction. All these natural and human factors could cause significant uncertainties which could not match with the condition required for those models. The existence of these uncertainties make a realistic mathematical models suitable to each airport very difficult if not impossible.

In fact, all the environmental problems associated with airport operations are related with the operation of air and surface traffic, and the same design of airport operation could be used to evaluate both operational requirements and environment requirements. An integrated 3D design facility will benefit the WHAT IF scenario analysis especially for those long term planning design such as sustainability analysis. With the increased awareness of the significance of environment to human being, a large amount of operational and environmental monitoring data have been produced in large airports. It provides a good foundation for realistic analysis of airport operation and its related environmental problems. In the same time, the development of artificial intelligence, especially soft computing technology, makes it possible to accommodate uncertainties into airport decision models.

1.2 Potential of GIS and soft computing in airport environment modelling

As is well known, both Geographic Information Systems (GIS) [198, 29] and air transport have experienced rapid developments in recent years. The improvements in computer hardware and software have contributed to the wide application and development of GIS; at the same time, socio-economic changes and technology development has helped contribute to a sharp increase in air transport use and demand.

As a powerful spatial database management and analysis tool, GIS has found an important role in surface transport and formed its own, widely recognised moniker – GIS-T [125]. GIS-T covers much of the broad scope of surface transport and provides an efficient medium for planning and evaluating the future generations of surface transport systems [158]. However, the application of GIS to air transport is not well established. There is research into and applications of GIS to airport operations, but most focus on the land side operation, such as pavement management [122], underground water investigations [149], crash risk distribution analysis [154], etc. Recently, some applications on air side operations have emerged, such as the Aircraft Noise Monitoring and Management System (ANMMS) developed by the Los Angeles Department of Airports [3].

Among the various commercial GIS software, many embed GIS-T functions into their new versions or as plug-in components, such as the network analysis component for ArcView [1] and a series of GIS-T products of GIS/Trans [69]. There is also specific GIS software for GIS-T functions, such as TransCAD [43] . However, these software are dedicated to surface transport, and do not consider the issues presented by, for example, air side operations. Furthermore, the fourth dimension – time – is generally not a part of most GIS software. Compared with surface transport, the application of GIS to air transport has not to date received sufficient attention from the GIS fraternity. This is surprising, since the aviation industry is developing at such a speed in recent years that its influence on the environment and society cannot be ignored. Today there are many computer models and database management systems applied in the daily operation of aviation, such as the airport operation model SIMMOD [23], the aircraft noise model, INM [41], and the emission model, EDMS [9]. In addition to these models, most relevant data such as flight trajectory and daily operations data can be found in the database systems of most major airports. The common feature of these models and data is the way in which they handle the spatial dimension, although they are processed by different systems for different purposes. With increasing demand for air travel and the environmental considerations which arise, the aviation industry faces huge challenges as a result of seeking higher efficiency on its operations and planning activities. This poses the requirement to integrate these different models and data to realise higher operating efficiency and support a more environmentally benign operation.

Clearly GIS is a suitable platform to integrate various models and data. With the development of the GIS application, the function of GIS has exceeded its original purpose [160]. From this point of view, GIS is the suitable interface for integrating different computer models with monitoring data, existing databases and users. However, compared to surface transport, air transport has its own special spatial and temporal characteristics. These include:

- A third dimension to handle flight trajectory is necessary;
- Heavily influenced by uncertain and dynamic weather conditions, especially wind direction;
- Environmental considerations, especially aircraft noise sensitivity;
- Significant impact on land use in the vicinity of an airport;
- Continuous expansion and modification of infrastructure;
- Uncertainties in 4D (3D+time) have to be considered.

These special features are not addressed directly in the present structure of GIS-T applications. The present GIS and GIS-T are designed to answer queries concerning 2D location only. However, queries on 3D and even 4D (time) are typical features of airport operation. A fully developed GIS-T has to meet many diverse needs including transportation inventory, modelling and operational problems, so a GIS application for air transport has to meet the diverse needs of air transport too. Therefore, it is necessary to investigate ways to apply GIS to the air transport system to accommodate these special requirements. Considering the next generation of air transport operations, where Global Positioning Systems, GIS and an Intelligent Air Transportation System will play the central role in its planning and operation. Here, the key technology in applying GIS to airport environment model is the realistic model established from real world data. Compared with theoretical models derived from assumptions and simplification of the real world, a realistic model has to consider the large amount of uncertainties existing both in the GIS representation and airport operations.

A large amount of airport operation data, such as weather conditions and human perception of the environment disturbance, are associated with uncertainties. In the same time, the spatial and temporal data in GIS have significant uncertainty as well. With the continuing development of Geographical Information Systems (GIS), the issue of data quality has become a major concern in its further application [61, 76, 75, 114]. Perfect data are rarely obtainable in the real world since most data contain errors and missing values. The traditional computing methods employed in GIS have difficulty in dealing with these kinds of imperfect data, and as such a methodology is required which allows for and that can handle imperfect data.

At the same time, airport development involves many human perception issues, such as noise disturbance and economy factors. Compared with other physical parameters, human perceptions are more difficult to model with an accurate measurement due to their very nature. The same thing may mean different things to different people. Therefore, technology in dealing with such uncertainty is an essential part of an airport environment evaluation model.

With the development of soft computing, such techniques are available to airport environment models now. Neural networks, fuzzy sets, rough sets and grey sets provide us with choices in dealing with uncertainty in airport environment evaluation. Neural networks are good at simulating a black box where input and output training data are available. Explicit knowledge about the black box is not required and they are capable of simulating very complicated relationships between inputs and outputs. The abundant in-situ data collected at airports provide ideal conditions for the application of neural networks. Fuzzy sets aim at representing those objects with fuzzy boundaries, and are suitable especially in representing human perception in fuzzy concepts such as sustainability. Rough sets approximate a set using two crisp sets, which provide tools for approximating influenced area. Grey sets appear as a new field, and provide opportunity to represent those vague objects in a map and preserve their uncertainty information. The uncertainty in airport environment models could be modeled using different models according to their different features, and a thorough study of the application of these modules would be of benefit to airport environment evaluation, especially to the sustainable development evaluation as well.

1.3 Overview of the dissertation

This thesis investigates the essential techniques in developing an airport environment evaluation model. The next chapter gives the background of airport environment models. Chapter 3 presents our methodology in dealing with uncertain information in the airport model, and then Chapter 4 discusses various algorithms in detail and Chapter 5 demonstrates some experiments for the proposed methods. While chapter 6 describes a prototype system for airport environment evaluation and decision support, and in the end we give our conclusions of the whole thesis.

Chapter 2

Background and related work

2.1 Airport environment problems

The environment by-product is becoming more serious than ever before with the development of aviation [133, 78]. The key for taking action in advance is to know what it would be in advance, and check the consequence of our planning operation before they are realized physically. In aviation, this function has been realized well for operation analysis, passenger forecasting, noise contour etc [95, 38, 182, 68, 12, 41, 14]. However, a synthesized audit of the potential environment limitation is still an open problem.

The environment problems had not been paid much attention in the early days of aviation, and a dramatic turnabout developed in 1960s with the increased awareness of the environment problems in general [14]. Because of the important role of the airport to the local economy, its propelling to the development of the economy is stressed far more than its negative influence to the environment. It is this kind of positive action to the economy that drives the aviation to expand at a surprisingly high speed. Whereas, unlike in its childhood when the aviation was so weak that it did not have any significant influence on its environment, current aviation has become an important cause for changes in the global environment. It is now at the stage when its environmental by-products can never be neglected again and its negative effect to the survival of humanity has to be taken account seriously. Compared with surface transport, environment pollution at airports shows not only 2D spatial feature, but 3D as well, as shown in Figure 2.1.

The environment problems caused by the operation of airports can be classi-

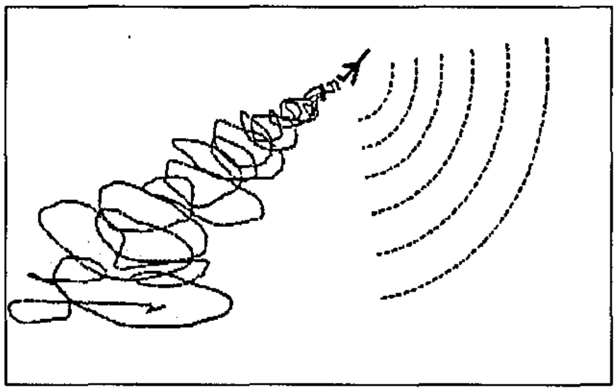


Figure 2.1: Pollution from 3D air traffic

fied as: noise, emission, waste, fuel consumption, water pollution as well as land use effect. Most of the environment researches for airport focus on the noise in the vicinity of the airport at this moment because of its direct disturbance to the local residence [41, 35, 55, 45, 66, 186, 188, 46, 119, 127]. The noise is produced by engines of the aircraft and influences the Land use near the airport seriously. With the development of new generation of quiet aircraft, the noise level is improved for the airport, but its frequency is becoming higher and higher as a result of satisfying the increased demand for air transport. Because of the simultaneous effect, most of the current complaints about airport environment come from the noise disturbance [45, 66, 188, 127], and the airport development is facing increasing pressure from their noise effect.

However, noise is only part of the environment problems in airports. It is highlighted because that it has the simultaneous effect compared with others. From the point of view of sustainable development, it may not be the most significant one in the environment problems. There may be some long term effects to the health of people, but its effect to the area would disappear when it stops. On the contrary, the influence of the gas emission and pollution in water system is far more complex than noise and has a long term effect. The noise would not accumulate with the long term operation of the airport, but the gas emission and water pollution do.

Trains and cars do produce some noise [107], but it is negligible compared to aircraft. The main source of noise from an airport is aircraft, and it disturbs only the close vicinity area around the airport. Unlike noise, the serious gas emissions from the airport are produced not only by aircraft, but also cars, buses and trains. The individual car has less emission than individual aircraft, but the huge amount of cars bring more emissions than aircraft as a whole. Furthermore, the emission from cars does not only influence the vicinity of the airport, but also other areas along the road networks. The aircraft would distribute emissions along its flight route, but its emissions would disperse in a large area. It does contribute to the global warming but does not influence the specific area as seriously as cars. Considering the different number of the moving cars and aircraft every day in the world and their different distribution space, it is obvious that the road traffic brings more serious gas emissions to the local people than air flight. Whereas, due to the interface and hub function of the airport, it concentrates every mode of transportation within a narrow area and then the combined gas emissions are more serious [144].

Some of the gas emissions could be removed from the airport area by wind and rain, but most of them would be left in the soil or underground water system. It seems not so serious to cause enough cautions now but it does not mean it would be the same in the future. In the same time, the de-icing process in the airport produces lots of chemical pollution and it would cause serious degradation to the local water and soil system if something went wrong [185]. The waste in the terminals and from various facilities in the airport would cause some problems too without compatible treatment.

All the noise, emission, soil and water pollution would change the land use dramatically [20, 22]. Those areas under the highest noise disturbance and serious emission influence would not be suitable for residence and the ecology system suffer very much too. Because of noise problem alone, many properties near to the airports have been devalued seriously [127]. On the other hand, this increased pollution comes as a result of consumption of fossil fuel, a very limited source for the human being [120]. An over consumption would threaten the life of the future generation.

Therefore, the development of airports brings a serious of environment problems, such as noise, gas emission, water pollution, waste, land use and fuel consumption problems. The extreme of any one of these problems would threaten the future life of human being as well as the ecology near the airports. With the development of the civilization of the whole society and the increased pressure from the surface transport, it is inevitable that more and more airports would appear everywhere. An efficient 3D multiple model transportation system is the future of the global transportation system. Therefore, it is not realizable that we limit the influence of the airports only to some desert area so as to avoid the annoying problem of environment. The only feasible way for the aviation is to develop in a self controlled way - so called sustainable transportation system [147].

The sustainable development requires that the airport satisfies not only the contemporary people but also their children, the future generation and forever. It is true that the gas emission and water pollution may not appear very serious for some airports today, but we may not say the same in the future. The environment degradation is not revertible in some sense, and it would be too late to recover when the gas emission and water pollution influence our life in the same strength like what noise does today. Aviation is a global industry, and it would be a global problem if something wrong and difficult for any individual to change it. Therefore, compared with the application of Information Technology (IT) in the surface environment analysis, the application of IT to aviation environment audit has more reasons to go far in advance of the development of the industry itself. This is determined by the audit itself: the early we find our potential problems, the better chance we have to avoid it ! Therefore, an airport oriented environment management decision support system is a necessary basis for the further development of aviation industry.

2.2 Sustainability of airport operation

With the increasing awareness of the environment problems, sustainability has become a very popular topic in various fields. Whereas, sustainability itself involves so much that it is difficult to give a very clear description. Generally, a sustainable development means the development should satisfy both current and future demand. Here, sustainability is related closely to the development, and hence it means a development with lower price rather than stop. This definition is a very general description and we have to find its concrete meaning for a special field. "A sustainable transportation system is defined as one in which fuel consumption, vehicle emissions, safety, congestion, and social and economic access are of such levels that they can be sustained into the indefinite future without causing great or irreparable harm to future generations of people throughout the world." [147] In fact, with the ongoing globalisation, it is inevitable that the demand for transport to increase become bigger and bigger [98]. The same situation can be found in airport development [78]. One side, the demand for additional capacity keeps increasing; on the other side, the environment consideration requires the control of its further expansion. This conflict is the centre content of the sustainability of airport development.

Unlike other transport modes, the airport serves not only for aviation but also involves surface transport, it is especially true for the big airport acting as a hub of the regional transport. The airport provides a transit station where various modes of transport meet and divert. In fact, the airport plays as a sink and source of the traffic along the surface road and airway networks. Thus it results a highly concentration of the various traffic in a relative small area. In the same time, with the development of the airport, a series of businesses and services related to the airport are introduced into this area, and the increased employment brings in a further increasing of its neighbour residence. All of these in turn rise the problem of sustainability for its further development. On the other hand, the individual airport is only a node in the whole networks, its position in the whole network need to be taken into account for its sustainability. Therefore, the sustainability of an airport can not be accessed only by itself, and it has to be related to all the partners. Thus, the sustainability of airport consists of network sustainability and local sustainability, as shown in Figure 2.2.

The sustainability of the airport means the ability of the airport to go on its development satisfying the demand of society without unrecoverable degradation to its environment. As the interface between different modes of traffic, the airports constitute a spatial network and its sustainability as an individual has to meet the sustainability of the network as a whole. In this sense, an unsustainable airport for the local region may be sustainable for the whole network. For example, a big airport appears unsustainable because of its heavy traffic, but the equivalent distributed small airports may bring more roads, occupy more lands, and disturb more natural environment. The big airport concentrates most air and surface transports and has less sustainability for its vicinity, but it does release other places to be disturbed and reduce the land use of road networks and runways. This is similar to the comparison between a big city and rural area. A big city concentrating various industries and traffics is less sustainable as an individual, but it may help the environment in larger area. The research has shown that rural area was less environmentally sustainable as a result of the extent of its dependency on the motor car [121]. Therefore, it is not appropriate to simply say that a big airport is less sustainable than small one. The conclusion depends on the scale of the consideration and concrete context of the specific airport. The conclusions may be on the contrary if the effect/passenger/km and the land use are taken into account. For example, if Heathrow airport is dispersed as 20 small airports, the effect/passenger/km and the land use would be far more inefficient than it is for Heathrow as a big airport.

Apart from the network effect, the sustainability of an airport is a kind of harmonisation of its partners within or near the airport, such as the airlines, surface traffic, relevant businesses, surrounding residencies, environment, ecosystem etc. This harmonisation requires that the interest of every partner should be respected and the benefit of one partner should not bring an unrecoverable destruction to another partner. Generally, the sustainability analysis involves three partners: society, environment and economy [31, 8, 145]. Whereas, the operation of the airport brings all the benefits and problems for the airport, and its surrounding ecosystem is the most vulnerable part in its partners, hence we

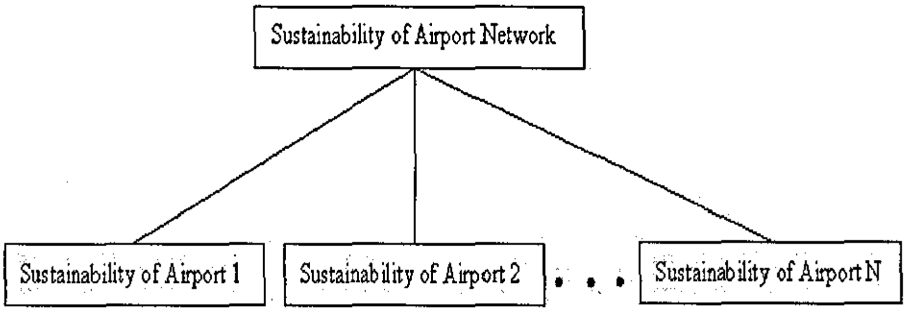

Sustainability of Airport Network
Sustainability of Airport 1
Sustainability of Airport 2
. . .
Sustainability of Airport N

Figure 2.2: The relationship between the individual airport and airport network

separate them from the three systems and then divide the airport as five sub systems: operation, environment, society, economy and ecosystem. The sustainability of the airport is determined by the incorporation of its five subsystems and its position in the network, as shown in Figure 2.3.

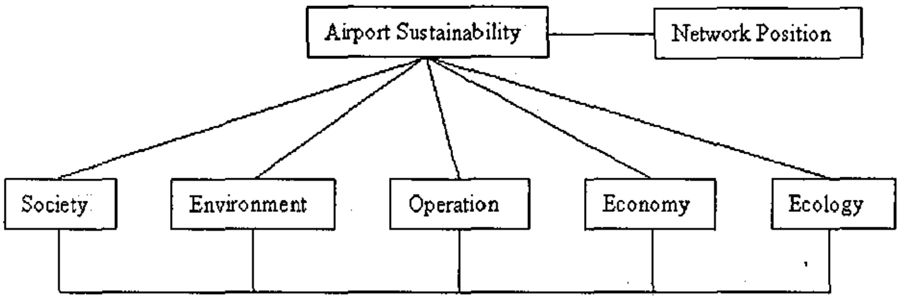


Figure 2.3: The airport sustainability and its factors

- The operation system is the special feature of the airport. Airport, especially those airports act as a hub of the local transportation system, forms a high density of air traffic and surface traffic in a relative small area. This include the landing and taking off of the aircraft, the movement of various vehicles in the taxi way and apron of the airport, the crowded traffic of those roads and railways leading to the airport. The change of the operation would bring a great difference to these traffic and then influence very much on the entire system. The operation system is the key system in the airport development.
- Environment system is the most typical indicator of the sustainability. For the airport, the environment system refers to its noise, gas emissions, waste, energy consume as well as water pollution. The existence of the airport would inevitably bring the environment changes to its local community, and the problem is what is the acceptable change. Most of the

current practices highlight the noise problems, but gas emission may be more serious if take into account their long term effect.

- Society system includes the residence of the neighbourhood area, the local council and the native government, various agencies like schools, medical centres and so on. The society is the consumer of the airport and also the victim for its negative effect. Therefore, The society system plays the key role in the further development of the airport, and they have to make choice between the benefit and the harm, especially the current benefit and the long term degradation. The airport development can not go over the limit of the law that the society imposes on the airport.
- Economy system is the driver of the development of the airport, and it means the economy benefit from its operation. This benefit is very attractive to the local economy and brings high standard life and employment. It is the economic benefits that bring and keep driving that airport going on its development. Therefore, the sustainable development has to find its role in the economy system, otherwise it is impossible to realise it.
- Ecosystem is the surrounding ecosystem, include various animals, plants and birds. Ecosystem itself may not influence the development of the airport, but their fate is associated with the living enviornment of human being, and their extinction would influence the life of human being.

The five systems have a very sophisticated interaction with each other and their synthetic development determines the sustainability of the system, as shown in Figure 2.4. Usually, the economy is the motivation of the airport, and hence the requirement for developing economy enables the airport operation. The operation of airport would bring the development of economy but cause environmental problems inducing social concerns about the development of airport, and then limit the operation of airport. The change in ecosystem would influence the life of people and then increase the pressure on operation. It is the interaction and incorporation between the five subsystems that determines the sustainability of airport in the end. The five systems form an organised airport system and the state of the whole airport system is controlled by the combination of its five subsystems. The system could be stable only when the combination could harmonise every subsystem and any action of a single subsystem without consideration of others would introduce abnormal interruption to the whole system and then introduce unsustainable development. This kind of unsustainable development has the feature that some subsystem is developed quickly on the basis of the destruction of other subsystem. This kind of development would result in the collapse of the whole system in the end although for some period some single system appears to be prospective. For example, the increasing of the operation in the airport would bring economic benefits to the society, but its unlimited development would close the airport itself in the end if it introduces an unacceptable environment to the society.

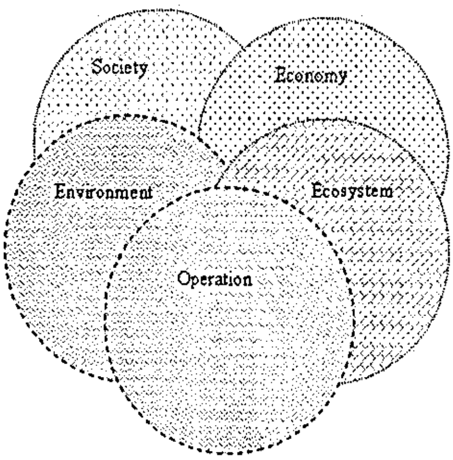


Figure 2.4: The subsystems of factors for sustainability

The sustainability of airport is easy to talk about with abstract language but difficult to access due to its complicated factors. First of all, the general concept of sustainability relates more to the entire society than to special individuals. It is easy to understand the sustainability of the human society, but very difficult to extract it from an individual airport. No matter what the individual airport does, its developing or closure hardly changes the sustainability of the whole society. However, it does matter if the same operation was carried out in all the airports and other sectors of our society. In this meaning, there is individual sustainability even if we stress the whole society or the global effect. It is the individual sustainability that constitutes the global sustainability. It is impossible for us to reach a sustainable development without starting from the individual partner. If all the partners reached their limit state, then the whole society reaches its limit without any doubt. Hence, the first step for us to carry out a sustainable development is to pursue the individual sustainable development. It is not a sufficient condition of global sustainability, but it is a necessary condition.

Sustainability is a general concept, and different people may have different interpretations. Therefore, indicators representing factors in different systems or subsystems have to be adopted to measure these factors quantitatively. An absolute input and output (waste output) measured increase in value for an airport can imply that the airport is moving away from environmental sustainability [187]. Obviously, the increase has to be measured against the specific context of the airport concerned. In this sense, the environmental sustainability of an airport is reflected by a balance between its environmental indicators and the environmental requirements of people. The balance has to be maintained continuously: the waste output (unwanted output) should never reach such a level that human beings cannot live with it. In this thesis, we adopt a narrow definition of sustainability: a continuous balance between conflicting indicators. Here, by conflicting indicators, we refer to those indicators with opposite effects to each other, such as the production of noise in the business of the airport and the human perception of noise by the surrounding population. A sustainable airport development requires a continuous balance between these conflicting indicators. This balance has to be maintained between each pair of conflicting indicators, and the sustainability of the airport is determined by the continuous balance of all involved indicators at different levels. In this thesis, we focus mainly on noise sustainability as a result of a continuous balance between noise and human perception, which is only part of the sustainability of an airport as a whole.

Another difficulty is the fuzziness of the sustainability [25]. In fact, what appears as a reasonable requirement now may not be acceptable for future generations. People always try their best to improve their life better and better, and the potential influence of our environment degradation to the future generation may not be well understood at this moment. Some factors seem acceptable today may not be the same for tomorrow. For example, most of the researches and practice to improve the environment in the airport are focusing on the noise problem [41, 65, 179, 46, 119, 127]. Whereas, the impact of noise can be detected now, but the potential change of the environment coming from the long term gas emissions has possibility to be more serious even if they seem not important now.

Therefore, the measurement of the sustainability of the individual partners of the whole society is meaningful even if it does not directly relate to the global sustainability, and the assessment of the sustainability has to consider the limitation of our knowledge and the complexity of the sustainability of airports. As aforementioned, the sustainability is influenced by many factors in different layers (network and individual airport). All these factors are dynamic and interactive, and then the assessment of sustainability should be dynamic and interactive. On the other hand, our limited knowledge prevents us from giving a very clear boundary between sustainable and unsustainable development, hence a fuzzy measure of the sustainability appears to be applicable.

Generally, the sustainability is a fuzzy concept. There is no very clear boundary between sustainable and unsustainable development. A development without any side product is the upper boundary of the sustainability, whereas, a development with a completely destruction of the surrounding environment and ecosystem could be regarded as the bottom boundary for the sustainability. Anything in between is a fuzzy concept between these two extremes. Therefore, the sustainability could be measured with a fuzzy membership rather than a concrete conclusion.

2.3 Sustainable airport development and environment models

The application of IT in aviation is very successful for its operation, and it is difficult to imagine what it would be without the modern IT technology in aviation. In this sense, the application of IT in aviation is more successful than other transportation modes in fact. For example, the air side operation model SIMMOD has been successfully applied in many airports in the world. With the development of the IT industry, more and more powerful operation models is appearing, such as the Airport Machine, TAAM, HERMES etc [132]. However, the environment audit for the airport has not reached the same level as operation. There have been some successful models for airport environment, such as INM, NOSIM [68, 41, 35, 132] and ADMS [33]. INM and NOSIM are noise models, which predict aircraft noise level according to a standard engine test curve. ADMS is an emission model for airport which is capable to give very detail about the possible concentration of the emission under given weather conditions. Odoni et. al. investigated existing models related to airport operation and reported their investigating result for models listed in Table 2.1 [132]. In their analysis, they classified the models into 5 categories according to their functionalities: capacity and delayed model, conflict detection and resolution models, human/automation models, cost/benefit analysis models and noise models. The distribution of the number of models in these five categories is shown in Figure 1. It is certain that the analysis made by Odoni et. al does not exclude the existence of other models, but it does reveal the general effort of the application of IT in airport environment management in comparison with other activities. Among these 5 groups, the first 4 groups are about operation simulation and analysis, which have always been in the focus of the aviation industry, and only the last one involves environment - noise. However, as aforementioned, noise is only a part of the indicators of environment, and there are many other relevant indicators which are not fully considered yet in these models. For these existing models, Odoni et. al. recognised their problems in communication: there are so many completely different formats for the same

Function	Models
CAPACITY AND DELAY MODELS	LMI Runway Capacity Model, FAA Airfield Capacity Model, AND (Approximate Network Delays), THE AIRPORT MACHINE, SIMMOD, TAAM (Total Airspace & Airport Modeller), HERMES (HEuristic Runway Movement Event Simulation), NASPAC, TMAC, FLOWSIM, ASCENT
CONFLICT DETECTION and RESOLUTION MODELS	RAMS (Reorganized ATC Mathematical Simulator), ARC2000 (Automatic Radar Control for the years beyond 2000), BDT (Banc De Test), NARSIM, ASIM (Airspace SIMulation), RATSG (Robust Air Traffic Situation Generator), TOPAZ (Traffic Organization and Perturbation AnalyZer)
HUMAN / AUTOMATION MODELS	SDAT (Sector Design Analysis Tool), DORATASK, MIDAS (Man-Machine Integration, Design, and Analysis System), PUMA (DRA)
Cost/Benefit Analysis Models	ACIM (The ASAC Air Carrier Investment Model), NARIM (The National Airspace Resource Investment Model)
NOISE Models	Integrated Noise Model (INM), NOISIM

Table 2.1: Existing computer models related with airport traffic management

airport operation, and their usage requires considerable training and expertise in the field [132].

Figure 2.5 reflects a general ratio between the different functionalities but does not exclude other existing models. There are a lot of similar models could be listed in Table 1. As environmental models, noise is the one who received the most investment in computer modelling for airport environment because of their obvious significance for the airport and residential community relationships. In addition to the models listed in Table 1, there are also some other environmental models developed in the recent years. For instance, the Heliport Noise Model (HNM) [57] developed by FAA, the Noise Integrated Routing System (NIRS) [124] developed by Metron, the Aircraft Noise Prediction Pro-

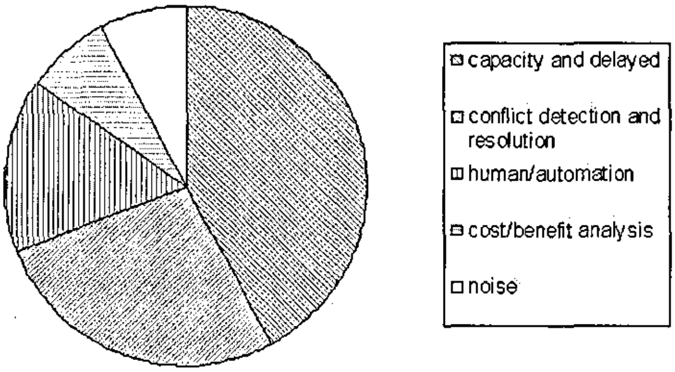


Figure 2.5: The distribution for the number of available models in Table 2.1

gram (ANOPP) [34] from NASA Langley Research Center, the Rotorcraft Noise Model (RNM) developed by Wyle Laboratories, Noise Model Simulation (NMSIM) also from Wyle [108]. In environment emission side, there is also a specific model for airport environment emission simulation, Emissions and Dispersion Modeling System (EDMS) developed by FAA [56].

As pointed by Odoni et. al, the formats of these models are very different from each other, and they separately analyse only some part indicators of the whole system. However, all these come from the same operation of the airports, hence there existed a huge amount of duplicated data inputs with different formats. Usually, these inputs require deep knowledge of expertise in airport operation. It limits the application convenience of computer modelling for airports, and makes it much more expensive than it should be, such as the complicated analysis of SIMMOD. In addition to this, the generalised equations adopted in these models can not fully reflect the specific features of individual airports due to their different geographical and operational environments. For instance, INM model relies on a predefined NPD (Noise vs. Power vs. Distance) curve, which is measured under specific weather situation and geographical conditions. Those conditions may not be the same for other airports, and the engine thrust and weather change are very difficult to know in advance, hence their results may not necessarily reflect the real situation around every airport. Figure 2.6 shows the monitoring data for departure BOE757 at Manchester airport, and it is obvious that its engine thrust is much more complicated than the theoretical data in NPD table.

From INM, the points in Figure 2.6 should distribute along a single curve with known engine thrust and distance. However, because of the uncertain wind

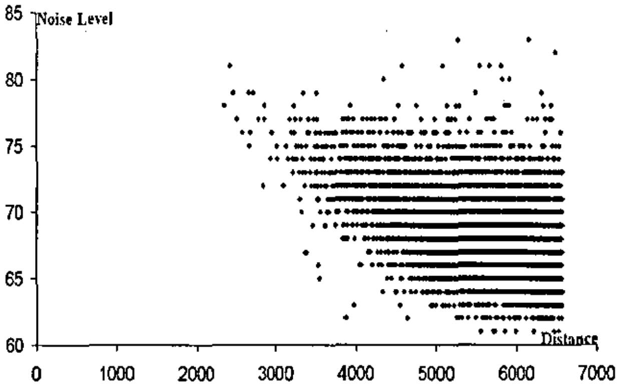


Figure 2.6: The monitoring data for BOE757 departing from Manchester airport

speed, temperature and takeoff weight, the aircraft speed does not follow exactly the scheme in INM, which results in a very different engine thrust configuration changing all the time according to wind speed and other factors. Obviously, INM model is acceptable as a simulation tool for general trends analysis, such as the average value in Figure 2.6, but not be able to reveal the real perception of airport noise for the residents. It is the individual pulse of noise that annoy people rather than the average value.

As a hub of the transportation network, operations at airports are typically spatial and temporal sensitive. Different flight trajectories and road networks will bring completely different distributions of noise and emissions at and around airports. Their operation times are also limited by the acceptability of the residents. Therefore, the computer models have to have spatial and temporal features, and provide a simple but powerful spatial and temporal design capability following the habit of air traffic control in the vicinity of airports. The linkage between Geographical Information Systems with noise and emission models have brought light to this field, but it is still an open field for the airport environment management software. An integrated model combining both the universal data format and uncertainty treatment is preferable in airport environment analysis.

Most environmental problems in the airports come from the routine operation of the airport, and the evaluation or audit of the potential environment problems has to be based on the operation analysis first. That means the different scenarios have to be simulated in the different systems again and again. Because of the possible changes of the planning outline and operation scenario, this audit would be terribly time consuming and high cost. On the other side, the existed models work well in general, but it may have bias for specific airports due to its general formulations. This kind of bias may not be important in the single analysis, but a combination of them from multiple models may enlarge it to an unacceptable level.

The environment audit serves for the evaluation of the potential environmental problems caused by the operation of the airport. Hence, it involves all the results of the models aforementioned. It has to be able to relate the sustainability of the airport to the components of the whole system. Obviously, all these require a model with feasibility to different data formats. On the other side, all the operations carried out in the airport are some kind of allocation problem with respect to the specific spatial and temporary requirement. Therefore, spatial and temporary are the two dominant characteristics of the airport operation. It indicates that a traditional database can not provide the necessary function in the spatial and temporary management of the airports. The spatial database GIS is a reasonable candidate for integrating the different data and requirements. Compared with other industry, the airports provide a better condition for the application of GIS in its environment auditing. With the increasing awareness of the potential environment problems and the social relationship with its neighbourhood, most big airports in the world have established an efficient environment monitoring system to record the noise, gas emission and other indicators. The operation of these systems had formed huge database on the environment data. In the same time, the rapid development of Internet and intranet enables the geographically distributed complaints and suggestions from the surrounding residents and societies to come into the airport in real time. On the other side, various data on operation are recorded in database for forecasting the future passenger demand. All these data are recorded in the corresponding airports and reflect their specific spatial and temporal conditions. Therefore, it provides a good foundation for the application of GIS system. Combined with the advanced technology in soft computing, GIS would provide a powerful tool in the airport sustainable development.

However, these data come from the real world measurement, and the complicated weather conditions and the engine response to these conditions will be combined into these data. It means that uncertainty becomes a significant factor in the environment prediction and analysis. The airport environment model has to be able to model those operations under uncertain and dynamic weather conditions and uncertain engine status. In addition to this, the spatial information itself is associated with uncertainty as well, such as the errors in measurement of spatial locations. More importantly, the sustainability is in fact a concept relating to human perception, which is much more uncertain than the physical measurement. For instance, for the same aircraft, some people may feel it's noise unacceptable, but some other people in the same location may not consider it a problem at all. A model tuned with data under a standard condition like INM is bound to produce big errors when the target airport has a complete different condition, and the uncertainty associated with each individual airport can only be tracked with data from the concerned airport. Therefore, the purpose of this thesis is to investigate the essential technique in dealing with the uncertainty associated with an airport environment evaluation model. We will focus mainly on noise but the same technology is applicable to emission and other environmental problems as well.

Chapter 3

Methodology

3.1 A Conceptual Framework for Society-Oriented Decision Support

Individual intelligence, like the individual expert in different fields, has attracted most efforts in research on Artificial Intelligence (AI). Most methodologies in AI are designed for simulation of the learning and reasoning process of the individual human brain, such as fuzzy logical reasoning [104], neural networks [89], genetic algorithms [71], etc. Multi-criteria decision making [117] is one of the areas in AI which considers more than one human brain, but it still does not consider fully the hierarchical interactions in human society. Social intelligence represents collective intelligence. Here, by society, we mean the hierarchical structure of human organisation. Social intelligence is therefore very different from individual intelligence in that individuals in a social environment cannot always optimise their own benefit, and where the benefit of the society as a whole may be more important. Therefore, a solution beneficial to individuals may not necessarily be an acceptable solution to a society. In this sense, social intelligence needs to involve interaction and negotiation - what has been called "communicative competence" [81]. Based on an investigation of the interactions between residents living adjacent to an airport and that airport's operations, we present a formulation of a hierarchical model of a society-oriented decision making process for airport development. The complexity of the interactions between the different individuals or agencies provides a challenge in simulating social intelligence. As a discussion for the further development of our research, we present our view on a social intelligence model based on what we have established in this research.

With the increasing application of information technology in airport operations and planning, recent years have seen the development of a number of models to help improve their operation, assess economic effects, evaluate environmental impacts, etc. However, there has yet to be developed a "society-oriented model" which has the potential to integrate all these sub models to form a hierarchical open structure to allow interaction and negotiation between different parties. Therefore, it is necessary to investigate the feasibility of such a model.

In most research into sustainability, the key subsystems are typically composed of local residents, the environment and the economy [31, 8]. For airports, their operation is often perceived to be the main cause of most sustainability problems - their ecosystem is more fragile and vulnerable. Considering these special features, the sustainability of an airport - and hence its future growth potential - has to consider the incorporation of and harmonisation between its operations, the environment, local residents, the economy and the ecosystem.

As a society-oriented model for the sustainability of airports, any model formulation has to deal with interactions and combinations between different subsystems. It is very difficult to establish a rule to say what is more important for the development of the airport. To those who see the economic subsystem of sustainable development as important, they may consider that further airport development should be given priority, but what of its impact on the other subsystems? At what price is this acceptable when its impact on other subsystems (e.g., the environment) are brought into the equation? For any given subsystem, there remain many components which need to be balanced before a decision can be reached. In the airport operation subsystem, for example, increasing capacity can be constrained by the facilities at the airport, such as length of the runway, terminal design, surface access transport and so on. For the environment subsystem, the status of the system is not only controlled by aircraft noise levels but also by the emissions arising from aircraft and ground operations as well as other factors which have to be taken into account. The local residents subsystem is more complex - people are very different from one another and what may be an acceptable level of airport operations for some may be completely unacceptable for others. The economic subsystem also involves very different components, such as the airport operator, airline operators, related businesses and so on. Their individual benefits may not be mutually beneficial. As for the ecosystem, it also has tremendous diversity, since different species of flora and fauna may have completely different adaptation capacities to the changes induced by the airport's operation. Therefore, when we view an airport in this context, the airport system consists of five subsystems, where each subsystem is composed of a series of special systems. This taxonomic chain could continue to a very fine level of detail depending on the level of analysis required. Hence, an audit of the sustainability of an airport has to reflect this special structure of the airport system.

It is well known that the decision making process for a very complicated problem in human society is, in general, not carried out by one person. A committee consisting of individuals with different expertise is a typical scheme. The advantage of this kind of process is that it engages different viewpoints, involves consideration of as many factors as possible and provides a more reliable solution after consulting all members. Here, we adopt a similar structure to simulate this kind of decision making process in our proposal for an intelligent audit system of airport sustainability. We call it the pyramid-committee structure. The "pyramid" represents the hierarchical/vertical structure of key relationships between the components and the system. The "committee" refers to the lateral/horizontal layers in the structure, which represent the relationships between different agents or "experts" in the same system, as shown in Figure 3.1. In this system, the process of assessing sustainability is composed of a series of actions to determine the sustainability of its components and then to negotiate their requirements at the committee level.

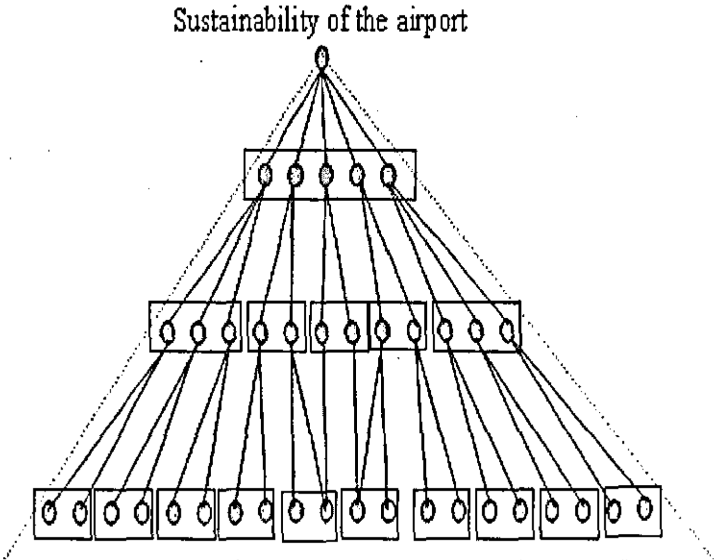


Figure 3.1: The structure of the pyramid-committee system

For the vertical structure, an intelligent audit agency sits at the top of the entire system, at the apex of the pyramid. Every subsystem is then represented by an agency in the first layer. For the final audit of sustainability, the apex is the "chair of the committee" of the first layer. Similarly, the five agencies in the first layer could be partitioned further into more subsystems. Thus, for example, all transport modes and their infrastructures could in principle be included within the subsystems. These would result in increasing the height of the pyramid because the whole system is formed layer by layer. Like the committee in the first layer, a series of committees in the sub-layers can be established. Each of these could be regarded as a professional committee in support of a specific area. Their chairs are then the agencies in the immediate upper layer. In this way, a hybrid system with a vertical pyramid and horizontal committee organisation structure is established to simulate a decision making system for resolving complicated problems involving both different interests and interest groups in a society.

Before any change in the airport's planned operation, the pyramid is by definition in a state of temporary stability. Any proposed change will break this stable state so the system has to be able to find a new state where each member can survive and the system returns to stability. Given the structure and function of the pyramid-committee system, its operation can be initiated by any node within the pyramid. For example, a proposal to improve the airport's sustainability overall can be initiated from the apex and then propagated to the nodes below or additional runway capacity can be proposed by any member of the operations committee for consideration and resolution. The specific requirement is discussed in the appropriate committee, and if its impact cannot be kept within that local committee, then the impact has to be evaluated by other committees and propagated in both directions to each relevant parent node. This process is shown in Figure 3.2.

The process of balancing each member's interest in a local committee is the basic unit of operation. Balance is in fact a process of negotiation of the impact of a proposal for change on each of the committee's other members. If the impact is deemed acceptable to each member and the change can be seen to benefit the committee as a whole, then this proposed change can progress. The whole process is a kind of iteration of the same operation for each committee. If a solution acceptable to each node is found, then this proposal for change succeeds, otherwise it fails. Obviously, the key here is the evaluation of the impact of this proposed change on other members or nodes. To initiate this interaction, we have first to establish their relationships. It could be established using the algorithms available in soft computing.

For interactions between the different components of this system, physical laws have the highest priority because of their sound foundations both in mathematics and in practice, such as the relationship between noise levels and attenuation. Expert knowledge and statistical relationships have the second priority in the system. Most of the relationships belong to this category because of the complexity of our understanding of what, for example, sustainability and

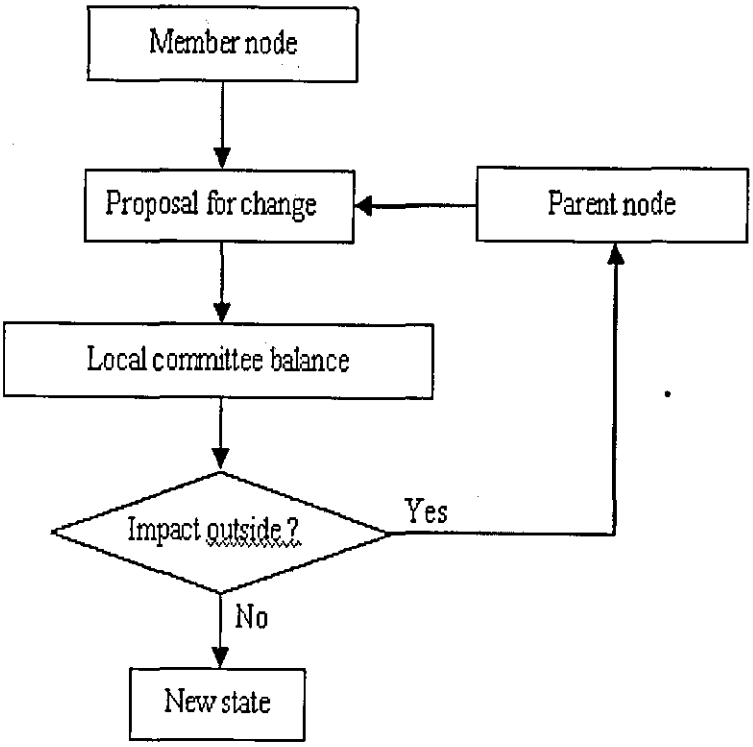


Figure 3.2: State transition operation

sustainable development "really means", together with the range of numerical indicators which are chosen to determine sustainability criteria. The relationships established from ANN could be applied only when others are not available. Similar to human emotions, intuition in this context is not necessarily as reliable as the application of logic but it is better than nothing.

In this way, a complicated interaction knowledge base is formed, and its links with the pyramid-committee are shown in Figure 3.3. The outcome of the committee deliberations in the pyramid-committee structure is highly reliant on the relationships in the knowledge base, and every "meeting" is in fact a negotiation process controlled by the relationships in the knowledge base. In terms of sustainability, any proposed development scenario has to meet the requirement of the co-existence of the different interests represented in the whole system. The "meta rule" illustrated in Figure 3.3 serves as the mechanism by which any conflict between different members in a committee are overcome with a multi-criteria decision strategy [103] where harmonisation is required and has to be made. By meta rule, we mean the rules that control the priority of rules in a knowledge base. For example, we have rules or maps in the knowledge base that establish the relationship between aircraft operations and the dispersion of emissions from those operations, as well as the relationship between aircraft operations and airline benefits. These rules can and often do conflict: for example, the enhanced benefits to an airline arising from increasing aircraft operations may mean a further degrading of local environmental conditions. However, a carefully planned and limited increase in aircraft operations may be more appropriate in order to satisfy passenger demand and airline competitiveness. In addition, environmental concerns may have overriding priority when aircraft emissions resulting from increased operations approach national limits or where internationally agreed standards may be breached. This example illustrates the point that these kind of "meta rules" cannot be considered in isolation or independently in such binary relationships - they have to be considered at a "committee" level.

With the pyramid-committee as aforementioned, a conceptual model system for evaluating sustainability and its associated indicators is presented and illustrated in Figure 6. It is obvious that the crucial part of the system is the knowledge extraction component which is at the centre and represents the basis of the whole system. By knowledge extraction, we mean the process to identify the relationships between different subsystems and factors. In the pyramid structure, we are emphasising the interaction between different nodes, such as the relationships between passenger numbers and emissions, between engine type and aircraft noise distribution, etc. The knowledge extraction process establishes the interactions between different relevant factors - for example, a

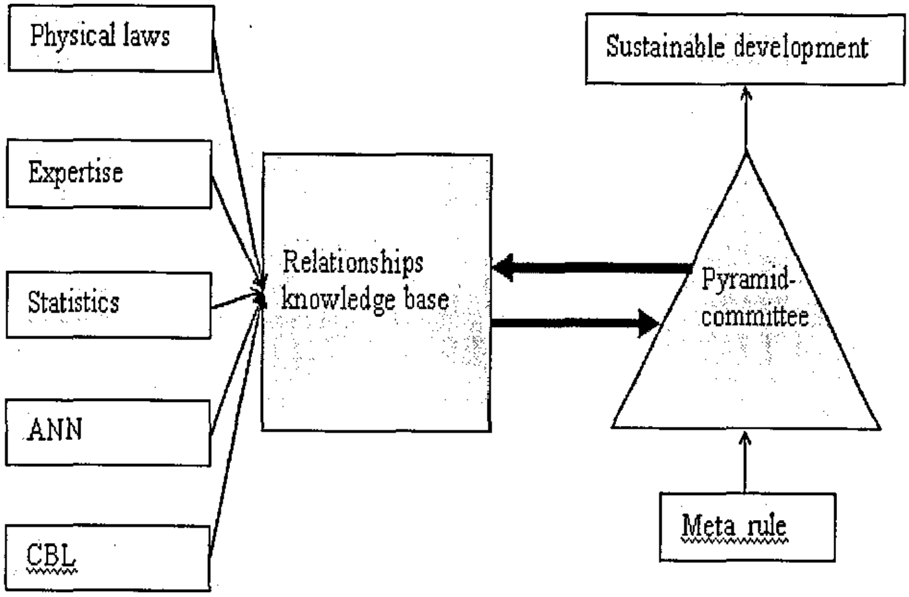


Figure 3.3: The links between the relational database and the pyramid-committee

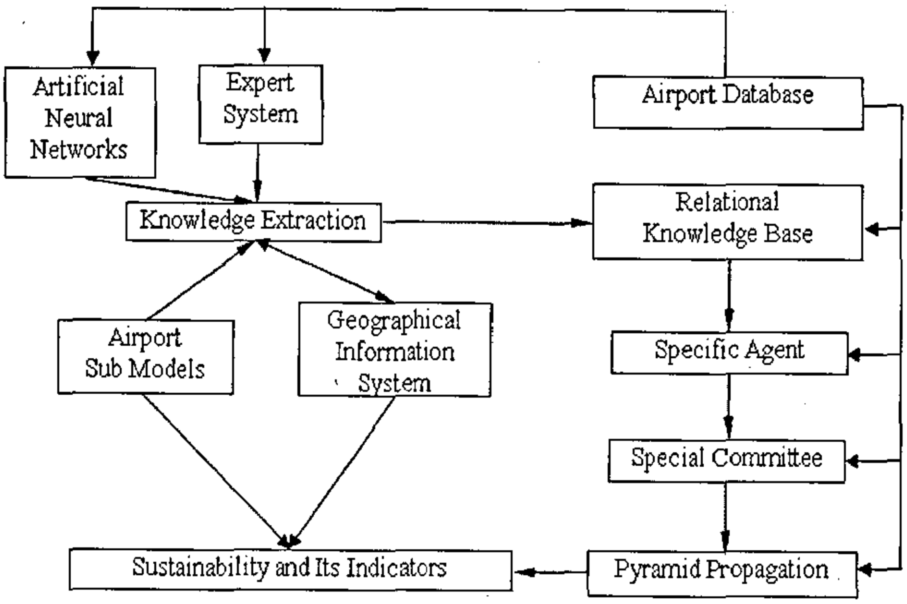


Figure 3.4: The structure of the intelligent audit system

mapping function between noise frequency and weather conditions, geographical location, aircraft speed and location is a result of the knowledge extraction process.

The airport database represents all the possible data about the airport in order to evaluate its sustainability, such as aircraft operation records, environment monitoring data, manufacturers' performance data on aircraft engines, cars and buses, etc. Based on these data, the knowledge extraction process is carried out to establish relationships between the different components of the system. The artificial neural networks, expert systems, geographical information systems, various airport operational and environmental models are potential servers for the knowledge extraction operation. With the relationships between different components of the system extracted, a relational knowledge base is formed which provides a necessary basis for the constitution of the pyramid-committee. According to the different subsystems of the airport, a pyramid scheme of committees simulating the human government inquiry process is established. Combined with the airport database and the extracted knowledge base, a hierarchical analysis of the sustainability of an airport can be conducted within the pyramid-committee.

In the final analysis, with the help of a geographical information system, the evaluation results are presented in spatial and temporal form since sustainability indicators in particular may have different spatial and temporal sensitivities. For example, local residents are more susceptible to the effects of aircraft noise events at night even if the levels may be lower than during daytime operations. Therefore, endurance to the same nuisance or pollutant can vary depending on where and when it happens. Therefore, a spatial and temporal distribution of these indicators in a GIS environment is more helpful for the user to evaluate the results.

Obviously, the most important technology in this pyramid system is the simulation of the interaction between different components of the system as well as the media to integrate these technologies. Here, we adopt neural networks to establish the interaction from in-situ data. The human perception is simulated using fuzzy sets. They are integrated into GIS and grey sets and rough sets are considered to represent those uncertainties associated with the spatial data in GIS. The same technology can also be applied to temporal data as well. Therefore, the key to the environment evaluation of airports is the way we apply soft computing into this domain. The rest of this chapter focuses on the methods available in soft computing.

3.2 Geographical Information Systems

The first obvious uncertainty associated with GIS is the spatial errors arising from the inclusion and use of imperfect data. There are many sources for spatial errors, such as positional accuracy problems with measurement, human interpretation problems resulting from different data providers and data integration problems as a result of multiple data sources [40]. These errors are not necessarily static, but may also be subject to change with time [29]. In addition, different operations of GIS may produce new errors related to the organisation of computer storage and data structure. Taking these errors together, the existence of spatial errors can seriously erode the quality of GIS output, especially when many outputs emanate from derived results rather than original measurement. However, most GIS functions simply assume perfect data quality in their operation, and as a result little attention has been paid to data quality issues [28]. Most efforts concerning data quality are still at the research level [61, 76, 75, 91, 114]. In the evaluation of spatial errors, statistical tests have been applied successfully to assess the significance of output errors [74, 28].

The second important uncertainty is human perception. The most important difference between human representation and machine representation of the world is the so called accuracy or completeness of the respective representations. In our daily communication, for example, it is rare for us to exchange precise or accurate numerical information. In most cases, we use a flexible language that employs loose concepts like "highly", "probably", "very possibly", etc. This kind of human language is not what the current generation of GIS can handle; rather we have to state clearly and in numerical terms what we mean by, for example, "near to a river" - i.e., 10.3 kms. This vagueness or fuzziness does not influence our communication, but it presents a big challenge for a machine to understand and represent. GIS represents geographical data as exact numbers, and hence its information retrieval is based on the traditional two-value logic: TRUE or FALSE. For instance, although aircraft noise around an airport is in fact a continuous distribution, we have to draw a line on a noise contour map somewhere to say that, on one side of this line, people exposed will be disturbed but, on the other side, they will not. This line can pass between two houses, with the implication that people can be classified as noise sufferers depending solely in which house they reside. This can mean that one household receives compensation (e.g., double glazed windows) whereas the other does not. This problem arises from the use of traditional two-value logic, there being no middle ground for a GIS based on this approach [29].

With the increasing use of GIS applications in industry and public life, the demands for decision support using spatial data are also increasing dramatically.

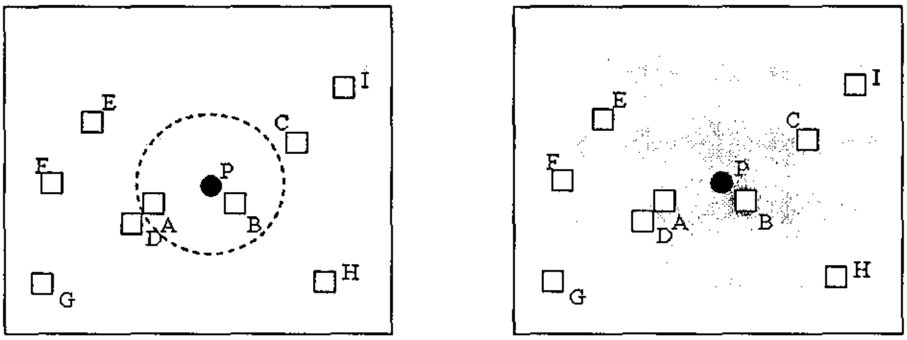
More and more decision support systems based on GIS are being developed and decision support functions are becoming embedded in GIS. In fact, the concept of GIS itself has also arguably changed, with decision support functions becoming an integral feature [111, 117]. To this end, GIS is no longer simply a special database for manipulating spatial data, but also a more fully functional information system with data analysis and decision supporting functions as well - for example, knowledge extraction or data mining functions are now seen to be a necessary part of any contemporary GIS [111, 84]. Such functions require capability in reasoning with available data, and an ability to mimic human intelligence in making a decision under given constraints. Here, uncertainties like missing values, incomplete or vague information, unknown mechanisms and factors are the main issues influencing the quality of outputs. Unlike geographical data, the spatial distribution of attributes in GIS is much more complicated, and it is controlled not only by their spatial location, but also involves many other known or unknown factors. For instance, the distribution of atmospheric emission pollutants is controlled not only by geographical features but also by wind direction, wind speed and ambient air temperature. Each of these factors involves a huge amount of uncertainty and are subject to change with time. Obviously, uncertainty is an inevitable characteristic of any GIS and a modern system must be able to deal with it in order to provide a more reliable and human friendly service.

3.3 Fuzzy sets

Before we discuss fuzzy sets, we should clarify first what is meant by a crisp set. By crisp sets, we mean the traditional sets where only two classes exist: TRUE or FALSE. For a subset S in a domain U=x1,x2,,xn, a crisp set has the following mapping function

$$\mu(x_i) \longrightarrow \{0,1\}$$

For example, we consider the spatial relationship of some entities with a pollution source P. We want to find all entities which are near to P. With crisp sets, we can only classify the entities into two classes: 'near' or 'not near'. Therefore, we have to clarify what is 'near' in the first place. In GIS, we can give a user defined distance d, and query all entities which have a distance from P less than d. The result is shown in Figure 3.5(a). With distance d, we can draw a circle around P: all those entities outside this hashed circle are not near to P, and all entities inside it are near to P. According to crisp sets, only A and B would be included in the output for entities near to P. However, the entity D is adjacent to A and it is really difficult to comprehend intuitively why they are treated so differently. In fact, they have only a very minor difference in their distances to P, and so it would be more reasonable to classify them together. This is a typical problem arising from the use of crisp sets, and so we need a different approach to recognise this kind of conceptual fuzziness.



(a) (b)

Figure 3.5: Crisp sets and fuzzy sets

3.3.1 Fuzzy sets

Fuzzy set theory was proposed originally by Zadeh [232]. Fuzzy sets are designed to provide a representation for those values in the middle ground between the two values of traditional sets. They are designed to represent values accurately rather than precisely. For the entities in Figure 3.5(b), their distances to P are indicators of their spatial relationship to P. Instead of having two classes 'near' and 'not near' to P, we can allow some entities to be near to some degree. For instance, entity G could be classified as 'definitely not near' P whereas B would be 'definitely near' to P, with other entities in between. Compared with traditional sets, the characteristic function can therefore take not only 0 and 1 values, but also a number between 0 and 1.

Formally, a fuzzy subset S in a domain U=x1,x2,,xn is defined as a set of ordered pairs

$$S = \{\langle x, \mu_S(x_i)\rangle : x_i \in U\}$$

where $\mu_S(x_i) \longrightarrow [0,1]$ is the membership function of S and is the grade of belongingness of x to S [232]. With fuzzy membership, a fuzzy concept can be represented as

$$S = \sum_{i=1}^{N} \mu_S(x_i)/x_i \tag{3.1}$$

The membership $\mu_S(x_i)$ provides a powerful tool in representing imprecise boundaries of the kind shown in Figure 3.5(b). We can define a membership function as a function of distances of an entity to P, reaching 1 when the entity is adjacent to P but 0 when it is far away from P. Then we get a continuous spatial distribution of membership as shown in Figure 3.5(b). It is clear that A and D have similar memberships in a fuzzy set. Therefore, we can say that B is the nearest entity to P, A and D are near to P to a high degree, but G has the lowest degree near to P. Using Equation 3.1, we can express the concept of 'near to' P as

$$S_{near} = \frac{0.9}{B} + \frac{0.55}{D} + \frac{0.5}{C} + \frac{0.3}{E} + \frac{0.2}{I} + \frac{0.2}{F} + \frac{0.2}{H} + \frac{0.1}{G}$$

Clearly, this equation gives a more complete description of the relationships between these entities and P. The output of a fuzzy operation in a GIS gives a result like Figure 3.5(b). Therefore, fuzzy sets provide us with a powerful tool to represent these kind of classification errors [40, 77] and imprecise boundaries [110].

In the representation and analysis of spatial error, statistical analysis remains the main tool used in GIS applications but fuzzy sets are gaining in significance. Their increasing application even resulted in a confusion between probability and fuzzy set interpretations [60, 61]. Although clearly different from the traditional two valued logic, fuzzy sets are not so sensitive to small spatial errors [91], and more suitable when representing ambiguous boundaries. Hence its application has been investigated for most GIS operations involving uncertainties, such as the reasoning process [80], viewshed operations [58, 59], information representation [193], object modelling [126, 44], map similarity [82], area calculation [64], information retrieval [141] and data integration [6].

3.3.2 Intuitionistic fuzzy sets and their distances

Fuzzy sets can be compared using their similarity or distance measures defined on their memberships. For instance, the distance between two fuzzy sets is defined using the distance between the memberships of their elements [99]. In the case of intuitionistic fuzzy sets, the membership is represented by two values rather than by a single number. Therefore, with intuitionistic fuzzy sets, more parameters have to be taken into consideration when measuring their distance. Atanassov [17] considers the distances between intuitionistic fuzzy sets as two dimensional and defined the two dimensional (2D) distances for intuitionistic fuzzy sets. Szmidt and Kacprzyk [168, 171, 175] proposed and applied the three dimensional (3D) distances for intuitionistic fuzzy sets. However, although the 3D distances are accepted as a correct representation, many people do not think it necessary. G. Deschrijver, C. Cornelis and E. E. Kerre proved in their paper [48] that the 2D Euclidean and Hamming distance and also their 3D counterparts given by Szmidt and Kacprzyk all generate the same topology. Therefore, the mainstream opinion about 3D distances is that it is useless and not necessary. This conclusion implies that the hesitation margins in intuitionistic fuzzy sets are redundant information as well. However, we would show here that this is not always true.

Distance is a measure of the similarity or difference between sets[86]. Therefore, for a selected distance measure, the relative order or 'spatial distribution' of sets is fixed with respect to a selected reference set. This is very important in real world application like information retrieval in databases, fuzzy number ranking [184], decision making and cased based reasoning [173, 174, 172, 170, 169, 167, 165, 164, 163, 162, 161, 112, 178]. The information retrieval could be conducted consistently only when we have fixed 'spatial distribution', and the query like 'find the 5 most similar sets with respect to set A' could be implemented using a distance measure. If the hesitation margin was really redundant and the 3D distances could be completely replaced by the 2D distances, then the order of their measuring result or 'spatial distribution' of the measured sets have to match each other in both 2D and 3D representation, in other wards, they should be consistent.

Interval-valued fuzzy sets [153] and intuitionistic fuzzy sets [15] are mathematically equivalent [50, 194, 30, 27, 49]. However, there is much debate in the fuzzy logic research community on the semantic differences between them [42]. Due to mathematical equivalence, we focus only on intuitionistic fuzzy sets and the result can be easily applied to interval-valued fuzzy sets as well.

Intuitionistic fuzzy sets were introduced by Atanassov [15]:

***Definition* 1 (Intuitionistic fuzzy sets)** *An intuitionistic fuzzy set A in X is given by*

$$A = \{\langle x, \mu_A(x), \nu_A(x)\rangle \,|\, x \in X\}$$

where

$$\mu_A : X \to [0,1] \;, \;\; \nu_A : X \to [0,1]$$

and

$$0 \le \mu_A(x) + \nu_A(x) \le 1 \quad \forall x \in X.$$

For each x, the numbers $\mu_A(x)$ and $\nu_A(x)$ are the degree of membership and degree of non-membership of x to A respectively.

Obviously, an intuitionistic fuzzy set becomes a fuzzy set when $\nu_A(x) = 1 - \mu_A(x)$. The distance between two fuzzy sets A and B is defined as [99]

- the Hamming distance $d_1(A,B)$

$$d_1(A,B) = \sum_{i=1}^{n} |\mu_A(x_i) - \mu_B(x_i)| \tag{3.2}$$

- the normalised Hamming distance $l_1(A,B)$

$$l_1(A,B) = \frac{1}{n}\sum_{i=1}^{n} |\mu_A(x_i) - \mu_B(x_i)| \tag{3.3}$$

- the Euclidean distance $e_1(A,B)$

$$e_1(A,B) = \sqrt{\sum_{i=1}^{n}(\mu_A(x_i) - \mu_B(x_i))^2} \tag{3.4}$$

- the normalised Euclidean distance $q_1(A,B)$

$$q_1(A,B) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\mu_A(x_i) - \mu_B(x_i))^2} \tag{3.5}$$

However, an intuitionistic fuzzy set is different from a fuzzy set when $\nu_A(x) \neq 1 - \mu_A(x)$. In these cases, an extra parameter has to be taken into account when working with intuitionistic fuzzy sets: the hesitancy degree $\tau_A(x)$ of x to A [15, 16, 17]

$$\tau_A(x) = 1 - \mu_A(x) - \nu_A(x)$$

The Hesitancy degree $\tau_A(x)$ is an indicator of the hesitation margin of the membership of element x to the intuitionistic fuzzy set A. It represents the amount of of lacking information in determining the membership of x to A.

Two different distances of intuitionistic fuzzy sets can be adopted. If only $\mu_A(x)$ and $\nu_A(x)$ are considered, a 2D distance [17] can be adopted. However, if the third parameter, i.e. the hesitancy degree, is taken into account then a 3D distance can be adopted.

For two intuitionistic fuzzy subsets A and B defined on a finite universe of discourse X, Atanassov defined the distance functions between two intuitionistic fuzzy sets in [17] as:

- the Hamming distance $d_2(A,B)$

$$d_2(A,B) = \frac{1}{2}\sum_{i=1}^{n}[|\mu_A(x_i) - \mu_B(x_i)| + |\nu_A(x_i) - \nu_B(x_i)|] \tag{3.6}$$

- the normalised Hamming distance $l_2(A, B)$

$$l_2(A, B) = \frac{1}{2n} \sum_{i=1}^{n} [|\mu_A(x_i) - \mu_B(x_i)| + |\nu_A(x_i) - \nu_B(x_i)|] \tag{3.7}$$

- the Euclidean distance $e_2(A, B)$

$$e_2(A, B) = \sqrt{\frac{1}{2} \sum_{i=1}^{n} [(\mu_A(x_i) - \mu_B(x_i))^2 + (\nu_A(x_i) - \nu_B(x_i))^2]} \tag{3.8}$$

- the normalised Euclidean distance $q_2(A, B)$

$$q_2(A, B) = \sqrt{\frac{1}{2n} \sum_{i=1}^{n} [(\mu_A(x_i) - \mu_B(x_i))^2 + (\nu_A(x_i) - \nu_B(x_i))^2]} \tag{3.9}$$

Szmidt and Kacprzyk [168] modified the above distances to include the third parameter $\tau_A(x)$ as follows:

- the Hamming distance $d_3(A, B)$

$$d_3(A, B) = \frac{1}{2} \sum_{i=1}^{n} [|\mu_A(x_i) - \mu_B(x_i)| + |\nu_A(x_i) - \nu_B(x_i)| + |\tau_A(x_i) - \tau_B(x_i)|] \tag{3.10}$$

- the normalised Hamming distance $l_3(A, B)$

$$l_3(A, B) = \frac{1}{2n} \sum_{i=1}^{n} [|\mu_A(x_i) - \mu_B(x_i)| + |\nu_A(x_i) - \nu_B(x_i)| + |\tau_A(x_i) - \tau_B(x_i)|] \tag{3.11}$$

- the Euclidean distance $e_3(A, B)$

$$e_3(A, B) = \sqrt{\frac{1}{2} \sum_{i=1}^{n} [(\mu_A(x_i) - \mu_B(x_i))^2 + (\nu_A(x_i) - \nu_B(x_i))^2 + (\tau_A(x_i) - \tau_B(x_i))^2]} \tag{3.12}$$

- the normalised Euclidean distance $q_3(A, B)$

$$q_3(A, B) = \sqrt{\frac{1}{2n} \sum_{i=1}^{n} [(\mu_A(x_i) - \mu_B(x_i))^2 + (\nu_A(x_i) - \nu_B(x_i))^2 + (\tau_A(x_i) - \tau_B(x_i))^2]} \tag{3.13}$$

Grzegorzewski in [79] argues that these 3D distances do not show better performance than their 2D counterparts, because the third parameter $\tau_A(x)$ can be expressed in terms of the other two. Within the 2D representation, Grzegorzewski recently proposed 2D Hausdorff distances for intuitionistic fuzzy sets [79].

3.4 Neural networks

Artificial neural networks (ANNs) are inspired by the mechanisms of the human brain when establishing interrelations between a variety of information sources. This refers to intuitive reasoning rather than the logical reasoning normally executed by machine. One of the most popular training schemes is the back-propagation (BP) network [151]. The back-propagation neural network architecture is a hierarchical design consisting of fully interconnected layers or rows of processing units (Figure 3.6). The interconnections are called weights and provide the means for ANN to save knowledge, the process of "learning". This process modifies the weights by incorporating the errors in the mapped output. Based on the calculation of error gradients, such errors are then back-propagated from the output neurons to all the hidden neurons; subsequently all the weights are adjusted with respect to the errors. The BP process is repeated until the error output has been reduced to a specified minimum value. The weights are then fixed and saved as a record of the knowledge pertaining to this system. Thus for a given input, an output is then associated with the fixed weight system.

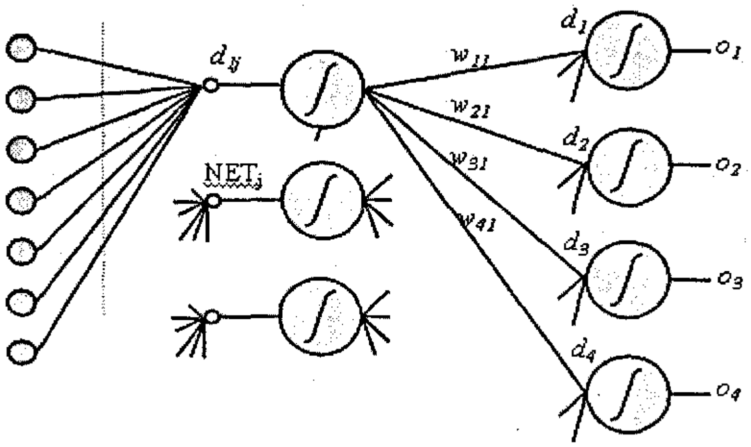


Figure 3.6: Structure of BP neural networks

The information processing operation facilitated by back-propagation performs an approximation of the bounded mapping function $f : A \mid \Re^n \rightarrow \Re^m$. This function is from a compact subset A of n-dimensional Euclidean space to a bounded subset f[A] of m-dimensional Euclidean space, by means of training with examples (x_1, y_1), (x_2, y_2), ..., (x_k, y_k), ... of the mapping, where $y_k = f(x_k)$. It is assumed that the mapping function f is generated by selecting xk vectors randomly from A in accordance with a fixed probability density function $P(x)$.

The operational use to which this network is put once training has been performed (on a set of experimental or observed data) makes use of the random selection of input vectors x in accordance with $P(x)$. ANN then models the mapping by utilising simple neurons based on either a linear or a non-linear activation function. Because of the large number of neuron connections, model behaviour is characterised by co-operation between neurons. Thus an incorrect decision introduced by a few neurons does not influence the outcomes from its associated mapping.

ANN has a robust quality with respect to uncertain or deficient information, even though such information influences many aspects of a complex system, for example the propagation of uncertainty. ANN can apply additional neurons and weights as required to take full account of such influences, and thus possesses an in-built capability for including any relation once it has been trained using a reference data set.

ANN may well have a large number of nodes, yet the activation function at each node is very simple. The complex knowledge contained within the training data is saved in the form of connections between the various nodes. The connections and activation functions determine the behaviour of the neural network. Thus, no matter how complicated the mechanisms, ANN has the capability of mapping it without having to incorporate a prior supposition or simplification.

The existence of large numbers of nodes needed to represent knowledge provides the robust structure for uncertain or incomplete inputs. The limited connection weights have the advantage of dramatically reducing the requirement for computer memory. Kolmogorov's "Mapping Neural Network Existence Theorem" [88, 90] has demonstrated that ANN is capable of implementing a mapping function to any desired degree of accuracy, and can thus enhance the decision support ability of GIS.

Neural networks are good at mapping a 'black box' between inputs and outputs. In environmental GIS applications, the spatial attributes only come from a limited number of monitoring stations over limited time intervals. Hence, it is necessary to obtain data for other spatial locations and time spans, and the only thing we can do here is to derive these data from monitoring stations near by or by monitoring over time. However, because of the complexity of the real world, there are no suitably precise mathematical models to simulate these data exactly. Therefore, neural networks provide a powerful tool for this type of situation, and it can also benefit a variety of decision making processes. Neural networks have mainly been applied to spatial interpolation [148], spatial attributes mapping [220, 219, 218, 217] and error simulation [26].

In addition to fuzzy sets and neural networks, there are other models which can be deployed in SC, including rough sets [138], grey systems [47] and genetic algorithms [71]. Fuzzy set and neural network models also have many extended versions, such as interval valued fuzzy sets [153], intuitionistic fuzzy sets [13], type 2 fuzzy sets [123], Hopfield neural networks [92], adaptive resonance theory neural networks [32], radial basis function neural networks [137] and fuzzy neural networks [102]. These methods are also receiving attention from the GIS community, and some have already been investigated, such as rough sets [5], genetic algorithms [195] and rough fuzzy sets [6].

3.5 Rough sets

The real world is inherently uncertain, imprecise and vague. Rough set theory focuses on the uncertainty caused by indiscernible elements with different values in decision attributes. It approximates the underlying set with two crisp sets. Therefore, the cardinality of elements in these two sets has a direct influence on the uncertainty of their corresponding rough set as a whole. Consequently, it is important that we consider the roughness of a rough set to have some understanding of the results in any decision making system. Knowing some bounds of this roughness before implementing the set operations can be important. Much research has been carried out on rough set theory, applications and their combination with fuzzy sets [51, 52, 53, 96, 97, 106, 130, 131, 143, 152, 196, 199, 230, 231]. As for roughness, there has been some research on the roughness of fuzzy sets [19, 24, 233].

3.5.1 Rough approximation and roughness

Rough sets consider any set as a set defined or described by a set of attributes. As pointed out by Pawlak [140](page 2),"Rough set philosophy is founded on the assumption that with every object of the universe of discourse we associate some information (data, knowledge)." Therefore, a rough set is defined with a set of attributes and the relation between these attributes. It is expressed as an information system where an information system is considered to be a data table with attributes as columns and objects as rows. Each entry in the table is a value for the information system. Here, we give a brief introduction to rough set concepts from the relevant literature.

***Definition* 2 (Information systems[106])** *We define an information system Λ by a pair (U, A), where U is a non-empty, finite set of objects called the universe and A is a non-empty, finite set of attributes*

$$\Lambda = (U, A).$$

Every attribute $a \in A$ of an object has a value. An attribute's value must be a member of the set V_a which is called the value set of attribute a.

$$a : U \rightarrow V_a$$

***Definition* 3 (Indiscernibililty relation[140])** *Given an information system (U, A) and subset $B \subseteq A$, B determines a binary relation $I(B)$ on U:*

$$(x, y) \in I(B) \text{ iff } a(x) = a(y) \text{ for every } a \in B$$

where $a(x)$ denotes the value of attribute a for element x.

Obviously, $I(B)$ is an equivalence relation, any two elements belong to $I(B)$ are identical from the point of view of a. An equivalence class of $I(B)$ is denoted by $B(x)$. If $(x, y) \in I(B)$, x and y are **B-indiscernible**[140]. Equivalence classes of $I(B)$ are called **B-granules**[140].

A decision system is a special case of an information system. Suppose d is a decision attribute and $d \notin A$. We have to make a decision for d based on the information in (U, A). Then an information system including d is a decision system

$$\Lambda = (U, A \cup \{d\})$$

where, d is called a decision attribute, and $a \in A$ as condition attribute. For a given information system, we can describe a set accurately with its uniquely identified attributes. However, it may happen that some elements with different decision attribute values may belong to a single $I(B)$. In this case, we describe the set using the notion of an approximation.

***Definition* 4 (Approximation[140])** $\Lambda = (U, A)$ *is a given information system, $X \subseteq U$ is a set. For a given set $B \subseteq A$, the set X is approximated with two sets $B_*(X)$ and $B^*(X)$*

$$B_*(X) = \bigcup_{x \in U} \{B(x) : B(x) \subseteq X\}$$

$$B^*(X) = \bigcup_{x \in U} \{B(x) : B(x) \cap X \neq \emptyset\}$$

here, $B(x)$ refers to an equivalence class of $I(B)$ containing x. $B_*(X)$ and $B^*(X)$ are called **B-lower** *and* **B-upper approximation** *of X, respectively.*

The B-lower approximation contains objects that are known to be members of X, it is the union of all B-granules that are included in the set. The objects in the set of the B-upper approximation are possible members of X. It is the union of all B-granules that have a nonempty intersection with the set. The B-boundary region is defined as the difference between the upper and the lower approximation[140]

$$BN_B(X) = B^*(X) - B_*(X)$$

This is the set of objects which have an unknown relationship with X. Some of them may be members of X, and others not. Obviously, elements in the B-lower set clearly belong to X, but elements in the B-boundary set may or may not belong to X. Therefore, the uncertainty of a rough set comes from the B-boundary set. The relative size of a B-boundary set with respect to the B-lower and B-upper sets has significant influence on the uncertainty of the set as a whole. In rough sets, the **accuracy of approximation** is defined to measure this significance.

***Definition* 5 (Accuracy of approximation[106])** $\Lambda = (U, A)$ *is an information system,* $X \subseteq U$ *and* $B \subseteq A$*,* $B_*(X)$ *and* $B^*(X)$ *are* B*-lower and* B*-upper approximation of* X *with respect to* B*. The accuracy of approximation is defined as* $\alpha_B(X)$

$$\alpha_B(X) = \frac{|B_*(X)|}{|B^*(X)|}$$

where $X \neq \emptyset$*,* $|B_*(X)|$ *and* $|B^*(X)|$ *are the cardinalities of* $B_*(X)$ *and* $B^*(X)$ *respectively.*

This coefficient is in fact very important, as it indicates how many elements in the set are certain and the accuracy of the approximation. The propagation of this coefficient under different set operations has not been fully investigated. It is also our view that the propagation of roughness of the set as a result of approximation operation has not gained enough attention. Here, we will focus on the propagation of this roughness.

Roughness [139] is a complementary concept to the accuracy of approximation. Roughness of a set X in information system $\Lambda = (U, A)$ is reflected by the ratio of the the number of objects in its B-boundary to that in its upper approximation.

***Definition* 6 (roughness of approximation)** *The roughness* $R^{\circ}{}_B(X)$ *for a set* X *approximated by* $B_*(X)$ *and* $B^*(X)$ *is defined as the significance of the uncertain elements to the set. This significance can be expressed as the ratio between the cardinalities of boundary set* $BN_B(X)$ *and* B*-Upper set* $B^*(X)$

$$R^{\circ}{}_B(X) = \frac{|BN_B(X)|}{|B^*(X)|} = \frac{|B^*(X)| - |B_*(X)|}{|B^*(X)|}$$

The roughness of approximation measure is, in some sense, the amount of uncertainty of the underlying set. A roughness of 1 shows that we have no certain knowledge on the underlying set, and a roughness of 0 means we know everything for sure about the underlying set. It is obvious that there is a relationship between the roughness of approximation $R^{\circ}{}_B(X)$ and accuracy of approximation $\alpha_B(X)$

$$R^{\circ}{}_B(X) = 1 - \alpha_B(X)$$

3.5.2 Set-oriented rough set interpretation and its operations

In this paper, similar to [128], we adopt the set-oriented interpretation of rough sets [96, 128, 138, 230] and define a rough set as a pair of disjoint sets.

***Definition** 7 [230] Let pair $apr = (U, B)$ be an approximation space on U and U/B denotes the set of all equivalence classes of B. The family of all definable sets in approximation space apr is denoted by $Def(apr)$. Given two subsets $A_*, A^* \in Def(apr)$ with $A_* \subseteq A^*$, the pair (A_*, A^*) is called a rough set.*

Here, A_* is the lower approximation of $X = (A_*, A^*)$, and A^* is the upper approximation of X. To differentiate the set-oriented rough set operation from others, we adopt A_* and A^* instead of B_* and B^* hereafter. The accuracy and roughness are also represented as α_A and R°_A. Here, we adopt the Iwinsky type set-oriented rough sets [96], although there are other similar definitions of set-oriented rough sets, such as the P-rough sets defined by Pawlak [138, 230].

Let U be the universe, and $X_1, X_2 \subset U$. The union, the intersection, the difference and the complement of rough sets have the following properties[96, 128, 230]:

- Union

$$A_*(X_1 \cup X_2) = A_*(X_1) \cup A_*(X_2), \quad A^*(X_1 \cup X_2) = A^*(X_1) \cup A^*(X_2)$$

- Intersection

$$A_*(X_1 \cap X_2) = A_*(X_1) \cap A_*(X_2), \quad A^*(X_1 \cap X_2) = A^*(X_1) \cap A^*(X_2)$$

- Difference

$$A_*(X_1 - X_2) = A_*(X_1) - A^*(X_2), \quad A^*(X_1 - X_2) = A^*(X_1) - A_*(X_2)$$

- Complement

$$A_*(\neg X_1) = \neg A^*(X_1), \quad A^*(\neg X_1) = \neg A_*(X_1)$$

It should be noted that different views exist for rough sets interpretation, so do their properties. A comprehensive review was given by Yao in [230]. Our approach follows the set-oriented view although future work will present our results for other interpretations.

3.6 Grey systems

The Grey System was proposed by Professor Julong Deng in 1982 [47]. Grey systems are concerned with the information belonging to the grey category. Because of insufficient information, most of the statistical characteristics of the system may not be clearly identified. However, the data available may reveal the range of information. We now provide a number of definitions.

***Definition* 8 (Grey numbers[113])** *A grey number is a number with clear upper and lower boundaries but which has an unknown position within the boundaries.*

A grey number $a^{\pm}$ for the system is expressed mathematically as [37]

$$a^{\pm} = [a^-, a^+] = \{a^- \leq t \leq a^+\}$$

where t is the unknown number represented by $a^{\pm}$, a^- and a^+ are the upper and lower limits of the unknown number. When $a^- = a^+$, we have a white number.

Each grey number is associated with a degree of greyness to represent its uncertainty.

***Definition* 9 (Degree of greyness for grey numbers[113])** *The significance of the unknown interval to the white number represented by a grey number is called the degree of greyness.*

The degree of greyness a function of the interval and the underlying white number. Because the underlying white number is unknown, the degree of greyness is usually expressed as a function of the two boundaries of a grey number.

$$g^{\circ}(a^{\pm}) = f(a^-, a^+)$$

In what follows, we will only focus on the application of grey numbers and the concept of degree of greyness to set uncertainty. There has not yet been a formal definition of grey sets although grey sets are referred as interval-valued fuzzy sets in some research [49]. We have defined grey sets specifically for grey numbers in a previous paper [214]. In this paper, we will define and analyse grey sets in a more general sense and we do not limit them to just grey numbers.

Chapter 4

Methodology extensions

4.1 Distance between intuitionistic fuzzy sets

4.1.1 Comparison of 2D and 3D distances

Because the hesitant degree can be expressed in terms of the membership and non-membership degrees, it is argued that 3D distances are not necessary as their 2D counterparts provide sufficient measures. As we will show, this is not the case, because 2D and 3D functions could lead to contradictory results. Therefore, we claim that 3D geometrical representation of intuitionistic fuzzy sets can not be simply replaced by their 2D counterparts.

To do this, in the following, the concept of consistency of distances of intuitionistic fuzzy sets is introduced. As aforementioned, the distance between sets is a measure of the similarity or difference between these sets. Therefore, two distance measures should give consistent results if one could be replaced by the other.

***Definition* 10 (Consistent distances)** *For any three intuitionistic fuzzy subsets A, B, C of the universe of discourse X, the distances D_1 and D_2 defined on X are said to be consistent if the following conditions hold:*

1. $D_1(A, C) = D_1(A, B) \Leftrightarrow D_2(A, C) = D_2(A, B)$
2. $D_1(A, C) > D_1(A, B) \Leftrightarrow D_2(A, C) > D_2(A, B)$

Clearly, two distances for intuitionistic fuzzy sets which are consistent maintain the same order between any triple of intuitionistic fuzzy sets. Therefore, when two distances are consistent then one of them can be replaced with the other, with the only effect on the magnitude of the distances but no change on the order between intuitionistic fuzzy sets. It is easy to prove that the 2D

(3D) Hamming distance is consistent with the 2D (3D) normalised Hamming distance, and the 2D (3D) Euclidean distance is consistent with the 2D (3D) normalised Euclidean distance. In particular, we have the following lemma for the 2D and 3D distances for fuzzy sets.

***Lemma* 1** *The 2D distances in Equations (3.6)–(3.9) and 3D distances in (3.10)–(3.13) coincide with their 1D distances counterparts in (3.2)–(3.5) when the two sets in comparison are fuzzy sets.*

Proof We will prove the result just for the Hamming distances d_1, d_2, d_3, because in similar way, we can prove it for the other distances.

For fuzzy sets $A, B \in U$, we have $\nu_A(x_i) = 1 - \mu_A(x_i)$, $\nu_B(x_i) = 1 - \mu_B(x_i)$ and $\tau_A(x_i) = \tau_B(x_i) = 0$. Thus

$$\begin{aligned} d_2(A,B) &= \frac{1}{2}\sum_{i=1}^{n}[|\mu_A(x_i) - \mu_B(x_i)| + |\nu_A(x_i) - \nu_B(x_i)|] \\ &= \sum_{i=1}^{n}|\mu_A(x_i) - \mu_B(x_i)| \\ &= d_1(A,B) \end{aligned}$$

and

$$\begin{aligned} d_3(A,B) &= \frac{1}{2}\sum_{i=1}^{n}[|\mu_A(x_i) - \mu_B(x_i)| + |\nu_A(x_i) - \nu_B(x_i)| + |\tau_A(x_i) - \tau_B(x_i)|] \\ &= \sum_{i=1}^{n}|\mu_A(x_i) - \mu_B(x_i)| \\ &= d_1(A,B) \end{aligned}$$

Therefore, we have

$$d_3(A,B) = d_2(A,B) = d_1(A,B)$$

Obviously, 2D and 3D distance representations are really redundant for fuzzy sets because they provide the same results than their 1D distance counterparts. Clearly, for fuzzy sets because the distances in Equations (3.6)–(3.9), (3.10)–(3.13) and (3.2)–(3.5) coincide with their counterparts we have that they are consistent.

***Corollary* 1** *The 2D distances in Equations (3.6)–(3.9) and 3D distances in (3.10)–(3.13) are consistent to their 1D counterparts in (3.2)–(3.5) for fuzzy sets.*

As aforementioned, it was argued that 3D distances on intuitionistic fuzzy sets were not necessary because their third parameter can be expressed in terms of the other two, and therefore the same results regarding the ordering of intuitionistic fuzzy sets would be obtained using their 2D counterparts. However, this is not the case as we show in the following:

***Lemma* 2** *The 2D distances for intuitionistic fuzzy sets in Equations (3.6)–(3.9) are not consistent with the 3D distances in Equations (3.10)–(3.13)*

Proof We provide the proof just for the Euclidean distance, the proof for the rest being similar.

Let $A = \{\langle x, 1-2v, v\rangle\}$, $B = \{\langle x, v, 1-2v\rangle\}$ and $C = \{\langle x, v, v\rangle\}$ be three intuitionistic fuzzy sets of the universe of discourse $X = \{x\}$ with $v \in [0, 0.5]$. According to Equation (3.8) and (3.12), we have

$$e_3(A,B) = |1-3v|, \quad e_3(A,C) = |1-3v|$$

$$e_2(A,B) = |1-3v|, \quad e_2(A,C) = \frac{\sqrt{2}}{2}|1-3v|$$

Therefore, we have $e_2(A,B) > e_2(A,C)$ when $e_3(A,B) = e_3(A,C)$, which obviously imply that e_2 and e_3 are not consistent.

As the above result shows, the application of a 2D and a 3D distance to the same set of three intuitionistic fuzzy sets provides a different ordering or representation of it. Using the 3D distance both B and C are at the same distance from A, while with the 2D distance B is further from A than C. Clearly, this last result is due to the fact that the hesitation margins of the intuitionistic fuzzy sets are not taken into account. Although the hesitation margin can be derived from the other two, this does not mean that it has not an effect on the representation of the intuitionistic fuzzy sets.

For the above example, the difference in the results obtained can be seen clearly when comparing both the 2D and 3D geometrical representation of the three intuitionistic fuzzy sets. The three intuitionistic fuzzy sets A, B and C are represented as points A, B and C in 3D interpretation and A_2, B_2 and C_2 in 2D interpretation, as shown in Figure 4.1.

In Figure 4.1, A_2B_2 is parallel to AB, hence its length is not changed. However, A_2C_2 has an angle with AC, and it is the projection of AC in plane $\mu\nu$, therefore, its length is less than AC. This is clearly a consequence of taking into account the third parameter of intuitionistic fuzzy sets. Although having the same relationship with the other two parameters in A, B, C, the effect of taking it into account does not lead to the same results regarding their relative ordering.

Another argument to support the use of the three dimensions of intuitionistic fuzzy sets when calculating their distance is the following. If only two parame-

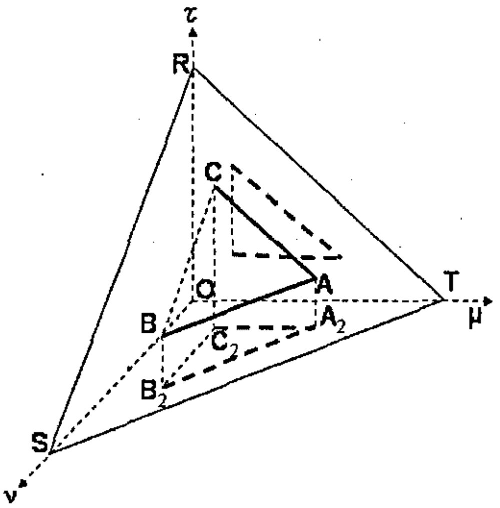


Figure 4.1: The inconsistency between 2D and 3D distances of intuitionistic fuzzy sets

ters among the three $(\mu(x), \nu(x), \tau(x))$ were sufficient to represent the distances because of their dependence $(\mu(x) + \nu(x) + \tau(x) = 1)$, then there would be no reason why the definition of the distance functions should based on $(\mu(x), \nu(x))$ and not on $(\mu(x), \tau(x))$ for example. As a consequence, the same results regarding the relative ordering should be obtained no matter which two parameters are used to measure the distance between intuitionistic fuzzy sets. Therefore, given three intuitionistic fuzzy sets their relative positions obtained with a 2D distance should be the same no matter we use $(\mu(x), \nu(x))$ or $(\mu(x), \tau(x))$. Again, this is not the case as we show using the same intuitionistic fuzzy sets A, B and C of Lemma 2. Using the distance e_2 we get opposite conclusions:

$e_2(A, B) > e_2(A, C)$ for 2D interpretation based on $\mu(x)$ and $\nu(x)$

$e_2(A, B) < e_2(A, C)$ for 2D interpretation based on $\mu(x)$ and $\tau(x)$

From the point of view of continuity or related concepts, the 3D distance does not reveal more than the 2D distance as proved in [48]. However, the application of intuitionistic fuzzy sets may require more than a distinction between two sets. The same continuity does not guarantee the same order, and hence the result for a query may be different. The 3D distance reveals the impact of hesitation margins in the relative order. This is an important factor in decision making because it reflects the influence of lacking of information [173, 174, 172, 170, 169, 167, 165, 164, 163, 162, 161, 112, 178]. A query based on 2D distances may not reflect the same situation as if based on 3D distances. It is necessary to keep

3D distances as a supplement to the simplicity of 2D distances and therefore it is worthwhile to investigate the 3D Hausdorff distances for intuitionistic fuzzy sets.

4.1.2 Extended Hausdorff distances between intuitionistic fuzzy sets

Given two intervals $U = [u_1, u_2]$ and $V = [v_1, v_2]$ of $\Re$, the Hausdorff metric is defined as [94]

$$M_h(U, V) = \max\{|u_1 - v_1|, |u_2 - v_2|\}$$

The Hausdorff metric applied to two intuitionistic fuzzy sets, $A(x) = [\mu_A(x), 1 - \nu_A(x)]$ and $B(x) = [\mu_B(x), 1 - \nu_B(x)]$, gives the following:

$$M_h(A(x), B(x)) = \max\{|\mu_A(x) - \mu_B(x)|, |\nu_A(x) - \nu_B(x)|\} \qquad (4.1)$$

The following 2D Hausdorff based distances between intuitionistic fuzzy sets have been proposed [94, 79]

- The Hamming distance $d_h(A, B)$

$$d_h(A, B) = \sum_{i=1}^{n} max\{|\mu_A(x_i) - \mu_B(x_i)|, |\nu_A(x_i) - \nu_B(x_i)|\} \qquad (4.2)$$

- The normalised Hamming distance $l_h(A, B)$

$$l_h(A, B) = \frac{1}{n}\sum_{i=1}^{n} max\{|\mu_A(x_i) - \mu_B(x_i)|, |\nu_A(x_i) - \nu_B(x_i)|\} \qquad (4.3)$$

- The Euclidean distance $e_h(A, B)$

$$e_h(A, B) = \sqrt{\sum_{i=1}^{n} max\{(\mu_A(x_i) - \mu_B(x_i))^2, (\nu_A(x_i) - \nu_B(x_i))^2\}} \qquad (4.4)$$

- The normalised Euclidean distance $q_h(A, B)$

$$q_h(A, B) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} max\{(\mu_A(x_i) - \mu_B(x_i))^2, (\nu_A(x_i) - \nu_B(x_i))^2\}} \qquad (4.5)$$

Obviously, distances (4.2)–(4.5) do not take into account the third parameter of intuitionistic fuzzy sets, $\tau_A(x)$ and $\tau_B(x)$. To do this, a 3D extended Hausdorff distance is necessary. A straightforward way to get this is to implement the corresponding difference between $\tau_A(x)$ and $\tau_B(x)$ in (4.1):

$$M_h(A(x), B(x)) = \max\{|\mu_A(x) - \mu_B(x)|, |\nu_A(x) - \nu_B(x)|, |\tau_A(x) - \tau_B(x)|\}$$

In the following we present the corresponding 3D extended versions of the above 2D Hausdorff based distances for intuitionistic fuzzy sets:

***Definition* 11 (Extended Hausdorff distances)** *For any two intuitionistic fuzzy subsets* $A = \{\langle x_i, \mu_A(x_i), \nu_A(x_i)\rangle : x_i \in X\}$ *and* $B = \{\langle x_i, \mu_B(x_i), \nu_B(x_i)\rangle : x_i \in X\}$ *of the universe of discourse* $X = \{x_1, x_2, \ldots, x_n\}$*, we have:*

- *The Hamming distance* $d_{eh}(A, B)$

$$d_{eh}(A,B) = \sum_{i=1}^{n} max\{|\mu_A(x_i) - \mu_B(x_i)|, |\nu_A(x_i) - \nu_B(x_i)|, |\tau_A(x_i) - \tau_B(x_i)|\} \tag{4.6}$$

- *The normalised Hamming distance* $l_{eh}(A, B)$

$$l_{eh}(A,B) = \frac{1}{n}\sum_{i=1}^{n} max\{|\mu_A(x_i) - \mu_B(x_i)|, |\nu_A(x_i) - \nu_B(x_i)|, |\tau_A(x_i) - \tau_B(x_i)|\} \tag{4.7}$$

- *The Euclidean distance* $e_{eh}(A, B)$

$$e_{eh}(A,B) = \sqrt{\sum_{i=1}^{n} max\{(\mu_A(x_i) - \mu_B(x_i))^2, (\nu_A(x_i) - \nu_B(x_i))^2, (\tau_A(x_i) - \tau_B(x_i))^2\}} \tag{4.8}$$

- *The normalised Euclidean distance* $q_{eh}(A, B)$

$$q_{eh}(A,B) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} max\{(\mu_A(x_i) - \mu_B(x_i))^2, (\nu_A(x_i) - \nu_B(x_i))^2, (\tau_A(x_i) - \tau_B(x_i))^2\}} \tag{4.9}$$

The following example shows that for the same set of intuitionistic fuzzy sets, opposite results can be derived when applying the above 3D extended Hausdorff distances and the corresponding 2D versions.

Example Let us consider the following three intuitionistic fuzzy sets A, B, C in $X = \{x\}$:

$$A = \{\langle x, 0.25, 0.25\rangle\},\ B = \{\langle x, 0.2, 0.2\rangle\},\ C = \{\langle x, 0.18, 0.32\rangle\}$$

The application of the 2D Hausdorff (4.2)–(4.5) results in

$$d_h(A,B)=0.05,\ l_h(A,B)=0.05,\ e_h(A,B)=0.05,\ q_h(A,B)=0.05$$

$$d_h(A,C)=0.07,\ l_h(A,C)=0.07,\ e_h(A,C)=0.07,\ q_h(A,C)=0.07$$

and we conclude that in any case B is closer to A than C. However, the application of the 3D extended Hausdorff distances (4.6)–(4.9) results in

$$d_{eh}(A,B)=0.1,\ l_{eh}(A,B)=0.1,\ e_{eh}(A,B)=0.1,\ q_{eh}(A,B)=0.1$$

$$d_{eh}(A,C)=0.07,\ l_{eh}(A,C)=0.07,\ e_{eh}(A,C)=0.07,\ q_{eh}(A,C)=0.07$$

and thus, we conclude that C is closer to A than B. It is clear that the application of the 2D Hausdorff distance and 3D extended Hausdorff distance may result in contradictory results. The reason behind this difference resides again in the implementation or not of the hesitation margins in the distance function. In 2D representation, the hesitation margins are neglected completely. However, $\tau_B(x)=0.6>\tau_A(x)=\tau_C(x)=0.5$ means that the real difference between A and B could be much greater than their 2D distances. Therefore, the 3D distance reveals the significant influence of the lacking of information in this comparison. It could be crucial in real world decision making where lacking of information cannot be simply ignored.

The above example proves the following result:

Lemma 3 *For intuitionistic fuzzy sets, the 2D Hausdorff distances are not consistent with the 3D extended Hausdorff distances.*

In the following we present some properties of the 3D extended Hausdorff distances shared by the 2D Hausdorff distances.

Lemma 4 *Let X denote a finite universe of discourse. All functions from definition 11 are metrics.*

Lemma 5 *For any two intuitionistic fuzzy subsets $A=\{\langle x_i,\mu_A(x_i),\nu_A(x_i)\rangle : x_i\in X\}$ and $B=\{\langle x_i,\mu_B(x_i),\nu_B(x_i)\rangle : x_i\in X\}$ of the universe of discourse $X=\{x_1,x_2,\ldots,x_n\}$, the following inequalities hold:*

$$d_2(A,B)\le d_h(A,B)\le d_{eh}(A,B)\ ;\quad l_2(A,B)\le l_h(A,B)\le l_{eh}(A,B)$$
$$e_2(A,B)\le e_h(A,B)\le e_{eh}(A,B)\ ;\quad q_2(A,B)\le q_h(A,B)\le q_{eh}(A,B)$$

Proof For any nonnegative numbers a,b,c, it is clear that $\max\{a,b,c\}\ge\max\{a,b\}$, hence $d_{eh}(A,B)\ge d_h(A,B)$. Obviously $\frac{1}{2}\sum_{i=1}^{n}[|\mu_A(x_i)-\mu_B(x_i)|+|\nu_A(x_i)-\nu_B(x_i)|]\le\sum_{i=1}^{n}\max\{|\mu_A(x_i)-\mu_B(x_i)|,|\nu_A(x_i)-\nu_B(x_i)|\}$. Therefore

$d_2(A,B) \leq d_h(A,B)$. Then we have $d_2(A,B) \leq d_h(A,B) \leq d_{eh}(A,B)$. The others inequalities can be proved in a similar way.

***Lemma* 6** *For any two intuitionistic fuzzy subsets* $A = \{\langle x_i, \mu_A(x_i), \nu_A(x_i)\rangle : x_i \in X\}$ *and* $B = \{\langle x_i, \mu_B(x_i), \nu_B(x_i)\rangle : x_i \in X\}$ *of the universe of discourse* $X = \{x_1, x_2, \ldots, x_n\}$, *the following inequalities hold:*

$$d_{eh}(A,B) \leq n, \quad l_{eh}(A,B) \leq 1, \quad e_{eh}(A,B) \leq \sqrt{n}, \quad q_{eh}(A,B) \leq 1$$

In contrast with conclusions from [79], we note that these inequalities do not include those 3D representations in Equations (3.10–3.13). The reason for this being that lemma 4 in [79] does not hold for some special cases. For example, when $A = < x, 0.7, 0.1 >$ and $B = < x, 0.3, 0.2 >$ then $e_h(A,B) = 0.4 > e_3(A,B) = 0.36$.

Lemma 5 presents a general relationship between 3D Hausdorff distances and 2D Hausdorff distances, but lacks to provide conditions to assure their consistency. The following results presents a condition under which both 2D Hausdorff distances and 3D extended Hausdorff distances are consistent, and therefore the same conclusions can be derived no matter which Hausdorff distance is used.

***Lemma* 7** *Given any two intuitionistic fuzzy subsets* $A = \{\langle x_i, \mu_A(x_i), \nu_A(x_i)\rangle : x_i \in X\}$ *and* $B = \{\langle x_i, \mu_B(x_i), \nu_B(x_i)\rangle : x_i \in X\}$ *of the universe of discourse* $X = \{x_1, x_2, \ldots, x_n\}$, *the following relationship holds between 3D extended Hausdorff distances and 2D Hausdorff distances: If* $(\mu_B(x_i) - \mu_A(x_i)) * (\nu_B(x_i) - \nu_A(x_i)) \leq 0$ *for each* $x_i \in X$, *then*

$$d_h(A,B) = d_{eh}(A,B),\ l_h(A,B) = l_{eh}(A,B),\ e_h(A,B) = e_{eh}(A,B),\ q_h(A,B) = q_{eh}(A,B)$$

Proof We only prove the results for the Hamming distance, because the other relationships can be proved in a similar way. When $(\mu_B(x_i) - \mu_A(x_i)) * (\nu_B(x_i) - \nu_A(x_i)) \leq 0$, $|\tau_A(x_i) - \tau_B(x_i)| \leq \max\{|\mu_A(x_i) - \mu_B(x_i)|, |\nu_A(x_i) - \nu_B(x_i)|\}$ holds. Therefore,

$$\begin{aligned} &\max\{|\mu_A(x_i) - \mu_B(x_i)|, |\nu_A(x_i) - \nu_B(x_i)|, |\tau_A(x_i) - \tau_B(x_i)|\} \\ &\qquad = \max\{|\mu_A(x_i) - \mu_B(x_i)|, |\nu_A(x_i) - \nu_B(x_i)|\} \qquad \forall x_i \in X \end{aligned}$$

Hence, $d_{eh}(A,B) = d_h(A,B)$.

The following result expresses the relationship between the 3D distances in Equations (3.10–3.13) and the 3D extended Hausdorff distances.

***Lemma* 8** *Given any two intuitionistic fuzzy subsets* $A = \{\langle x_i, \mu_A(x_i), \nu_A(x_i)\rangle : x_i \in X\}$ *and* $B = \{\langle x_i, \mu_B(x_i), \nu_B(x_i)\rangle : x_i \in X\}$ *of the universe of discourse*

$X = \{x_1, x_2, \ldots, x_n\}$, *the following relationships hold:*

$$d_{eh}(A,B) = d_3(A,B),\ \ l_{eh}(A,B) = l_3(A,B),\ \ e_{eh}(A,B) \geq e_3(A,B),\ \ q_{eh}(A,B) \geq q_3(A,B)$$

Proof

$$\frac{1}{2}[|\mu_A(x_i) - \mu_B(x_i)| + |\nu_A(x_i) - \nu_B(x_i)| + |\tau_A(x_i) - \tau_B(x_i)|] =$$

$$\begin{cases} |\mu_A(x_i) - \mu_B(x_i)| + |\nu_A(x_i) - \nu_B(x_i)| & \text{if } (\mu_B(x_i) - \mu_A(x_i)) * (\nu_B(x_i) - \nu_A(x_i)) \geq 0; \\ |\mu_A(x_i) - \mu_B(x_i)| & \text{if } (\mu_B(x_i) - \mu_A(x_i)) * (\nu_B(x_i) - \nu_A(x_i)) < 0 \\ & \text{and } |\mu_B(x_i) - \mu_A(x_i)| \geq |\nu_B(x_i) - \nu_A(x_i)|; \\ |\nu_A(x_i) - \nu_B(x_i)| & \text{if } (\mu_B(x_i) - \mu_A(x_i)) * (\nu_B(x_i) - \nu_A(x_i)) < 0 \\ & \text{and } |\mu_B(x_i) - \mu_A(x_i)| < |\nu_B(x_i) - \nu_A(x_i)|. \end{cases}$$

Hence, we can conclude that $\frac{1}{2}[|\mu_A(x_i) - \mu_B(x_i)| + |\nu_A(x_i) - \nu_B(x_i)| + |\tau_A(x_i) - \tau_B(x_i)|]$ is equal to the maximum of $|\mu_A(x_i) - \mu_B(x_i)|$, $|\nu_A(x_i) - \nu_B(x_i)|$ and $|\tau_A(x_i) - \tau_B(x_i)|$, and therefore $d_3(A,B) = d_{eh}(A,B)$. The same result can be obtained for l_3 and l_{eh}. For the Euclidean distance, we get:

$$\max\{(\mu_A(x_i) - \mu_B(x_i))^2, (\nu_A(x_i) - \nu_B(x_i))^2, (\tau_A(x_i) - \tau_B(x_i))^2\} =$$

$$\begin{cases} (\mu_A(x_i) - \mu_B(x_i) + \nu_A(x_i) - \nu_B(x_i))^2 & \text{if } (\mu_B(x_i) - \mu_A(x_i)) * (\nu_B(x_i) - \nu_A(x_i)) > 0; \\ \max\{(\mu_A(x_i) - \mu_B(x_i))^2, (\nu_A(x_i) - \nu_B(x_i))^2\} & \\ & \text{if } (\mu_B(x_i) - \mu_A(x_i)) * (\nu_B(x_i) - \nu_A(x_i)) \leq 0. \end{cases}$$

and we can conclude that $e_{eh}(A,B) \geq e_3(A,B)$. The same result can be obtained for q_3 and q_{eh}.

4.1.3 3D Spherical distance

Limitation of linear distances

Research in cognition science [39] has shown that people are faster at identifying an object that is significantly different from other objects than at identifying an object similar to others. The semantic distance between objects plays a significant role in the performance of these comparisons [190]. For the concepts represented by fuzzy sets and intuitionistic fuzzy sets, an element with full membership (non-membership) is usually much easier to be determined because of its categorical difference from other elements. This requires the distance between intuitionistic fuzzy sets or fuzzy sets to reflect the semantic context of where the membership/non-membership values are, rather than a simple relative difference between them.

In contrast to traditional fuzzy sets where only a single number is used to represent membership degree, more parameters are needed for intuitionistic fuzzy sets. Geometrical interpretations have been associated with these parameters, which are especially useful when studying the similarity or distance between intuitionistic fuzzy sets. One of these geometrical interpretations was given by Atanassov in [17], as shown in Figure 4.2(a), where a universe U and subset OST in the Euclidean plane with Cartesian coordinates are represented. According to this interpretation, given an intuitionistic fuzzy set A, a function f_A from U to OST can be constructed, such that if $u \in U$, then

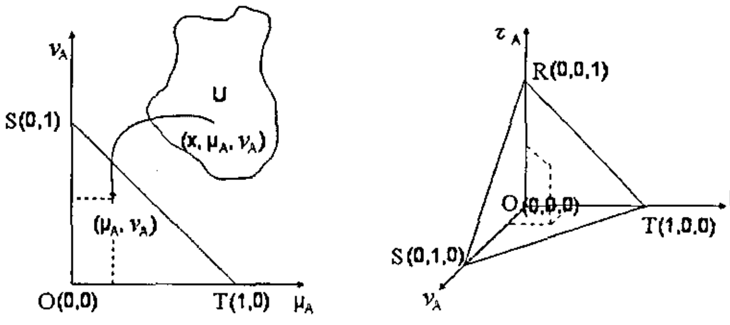
$$P = f_A(u) \in OST$$

is the point with coordinates $\langle \mu_A(u), \nu_A(u) \rangle$ for which $0 \leq \mu_A(u) \leq 1, 0 \leq \nu_A(u) \leq 1$, $0 \leq \mu_A(u) + \nu_A(u) \leq 1$.

We note that the triangle OST in fig. 4.2 (a) is an orthogonal projection of the 3D representation proposed by Szmidt and Kacprzyk [168], as shown in fig. 4.2 (b). In this representation, in addition to $\mu_A(u)$ and $\nu_A(u)$, a third dimension is present, $\tau_A(u) = 1 - \mu_A(u) - \nu_A(u)$. Because $\mu_A(u) + \nu_A(u) + \tau_A(u) = 1$, the restricted plane RST can be interpreted as the 3D counterpart of an intuitionistic fuzzy set. Therefore, in a similar way to Atanassov procedure, for an intuitionistic fuzzy set A a function f_A from U to RST can be constructed, in such a way that given $u \in U$, then

$$S = f_A(u) \in RST$$

has coordinates $\langle \mu_A(u), \nu_A(u), \tau_A(u) \rangle$ for which $0 \leq \mu_A(u) \leq 1$, $0 \leq \nu_A(u) \leq 1$, $0 \leq \tau_A(u) \leq 1$ and $\mu_A(u) + \nu_A(u) + \tau_A(u) = 1$.



a. 2D representation **b. 3D representation**

Figure 4.2: 2D and 3D representation of intuitionistic fuzzy sets

Most existing distances based on the linear representation of intuitionistic fuzzy sets are linear in nature, in the sense of being based on the relative difference between membership degrees [99, 15, 16, 17, 168, 166, 161, 176, 162, 177, 163].

All the above distances clearly adopt a linear plane interpretation, and therefore they reflect only the relative differences between the memberships, non-memberships and hesitancy degrees of intuitionistic fuzzy sets. The following lemma proves that:

Lemma 9 *For any four intuitionistic fuzzy subsets* $A = \{\langle u_i, \mu_A(u_i), \nu_A(u_i)\rangle : u_i \in U\}$, $B = \{\langle u_i, \mu_B(u_i), \nu_B(u_i)\rangle : u_i \in U\}$, $C = \{\langle u_i, \mu_C(u_i), \nu_C(u_i)\rangle : u_i \in U\}$ *and* $G = \{\langle u_i, \mu_G(u_i), \nu_G(u_i)\rangle : u_i \in U\}$ *of the universe of discourse* $U = \{u_1, u_2, \ldots, u_n\}$, *if the following conditions hold*

$$\mu_A(u_i) - \mu_B(u_i) = \mu_C(u_i) - \mu_G(u_i)$$

$$\nu_A(u_i) - \nu_B(u_i) = \nu_C(u_i) - \nu_G(u_i)$$

then

$$D(A, B) = D(C, G)$$

being D any of the above Atanassov's 2D or Szmidt and Kacprzyk's 3D distance functions.

Proof The proof is obvious for Atanassov's 2D distances. For Szmidt and Kacprzyk's 3D distances, the proof follows from the fact that

$$[\mu_A(u_i) - \mu_B(u_i) = \mu_C(u_i) - \mu_G(u_i)] \wedge [\nu_A(u_i) - \nu_B(u_i) = \nu_C(u_i) - \nu_G(u_i)]$$

imply that

$$\tau_A(u_i) - \tau_B(u_i) = \tau_C(u_i) - \tau_G(u_i)$$

If $|\tau_A(u_i) - \tau_B(u_i)| = |\tau_C(u_i) - \tau_G(u_i)|$, the condition in lemma (9) can be generalised to

$$|\mu_A(u_i) - \mu_B(u_i)| = |\mu_C(u_i) - \mu_G(u_i)|$$

$$|\nu_A(u_i) - \nu_B(u_i)| = |\nu_C(u_i) - \nu_G(u_i)|$$

This means that if we move both sets in the space shown in Figure 4.2(b) with the same changes in membership, non-membership and hesitancy degrees, then we obtain exactly the same distance between the two fuzzy sets. This linear feature of the above distances may not be adequate in some cases, because human perception is not necessarily always linear.

For example, we can classify the human behaviour as perfect, good, acceptable, poor and worst. Using fuzzy sets, we can assign their fuzzy membership as 1, 0.75, 0.5, 0.25 and 0. To find out if someone's behaviour is perfect or not, we only need to check if there is anything wrong with them. However, to differentiate good from acceptable, we have to count their positive and negative points. Obviously, the semantic distance between perfect and good should be greater than the semantic distance between good and acceptable. This semantic difference is not captured by using a linear distance between their memberships.

Therefore, a non-linear representation of the distance between two intuitionistic fuzzy sets may benefit the representative power of intuitionistic fuzzy sets. Although, non-linearity could be modelled by using many different expressions, we will consider and use a simple one to model it. Here, we propose a new geometrical interpretation of intuitionistic fuzzy sets in 3D space using a restricted spherical surface. This new representation provides a convenient and also simple non-linear measure of the distance between two intuitionistic fuzzy sets.

Spherical Interpretation of Intuitionistic Fuzzy Sets: Spherical Distance

Let $A = \{\langle u, \mu_A(u), \nu_A(u)\rangle : u \in U\}$ be an intuitionistic fuzzy set. We have

$$\mu_A(u) + \nu_A(u) + \tau_A(u) = 1$$

which can be equivalently transformed to

$$x^2 + y^2 + z^2 = 1$$

with

$$x^2 = \mu_A(u), \quad y^2 = \nu_A(u), \quad z^2 = \tau_A(x)$$

It is obvious that we could have other transformations satisfying the same function. However, as shown in the existing distances, there is no special reason to discriminate $\mu_A(u)$, $\nu_A(u)$ and $\tau_A(u)$. Therefore, a simple non-linear transformation to the unit sphere is selected here.

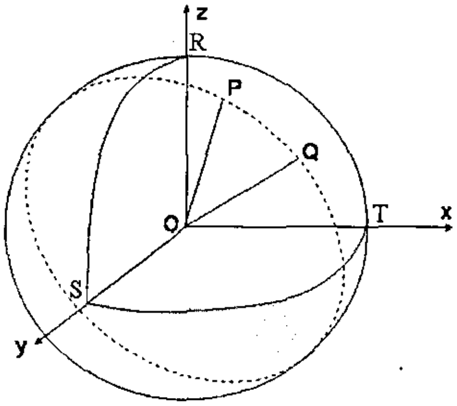


Figure 4.3: 3D sphere representation of intuitionistic fuzzy sets

This last equality represents a unit sphere in a 3D Euclidean space as shown in Figure 4.3. This allows us to interpret an intuitionistic fuzzy set as a restricted spherical surface. An immediate consequence of this interpretation is that the distance between two elements in an intuitionistic fuzzy set can be defined as the spherical distance between their corresponding points on its restricted spherical surface representation. This distance is defined as the shortest path between the two points, i.e. the length of the arc of the great circle passing through both points. For points P and Q in Figure 4.3, their spherical distance is [2]:

$$d_s(P,Q) = \arccos\left\{1 - \frac{1}{2}\left[(x_P - x_Q)^2 + (y_P - y_Q)^2 + (z_P - z_Q)^2\right]\right\}$$

This expression can be used to obtain the expression of the spherical distance between two intuitionistic fuzzy sets, $A = \{\langle u_i, \mu_A(u_i), \nu_A(u_i)\rangle : u_i \in U\}$ and $B = \{\langle u_i, \mu_B(u_i), \nu_B(u_i)\rangle : u_i \in U\}$ of the universe of discourse $U = \{u_1, u_2, \ldots, u_n\}$, as follows:

$$\begin{aligned} d_s(A,B) = \frac{2}{\pi}\sum_{i=1}^{n} \arccos\Big\{1 - \frac{1}{2}\Big[\left(\sqrt{\mu_A(u_i)} - \sqrt{\mu_B(u_i)}\right)^2 \\ + \left(\sqrt{\nu_A(u_i)} - \sqrt{\nu_B(u_i)}\right)^2 + \left(\sqrt{\tau_A(u_i)} - \sqrt{\tau_B(u_i)}\right)^2\Big]\Big\} \end{aligned} \tag{4.10}$$

where the factor $\frac{2}{\pi}$ is introduced to get distance values in the range $[0,1]$ instead of $[0, \frac{\pi}{2}]$. Because $\mu_A(u_i)+\nu_A(u_i)+\tau_A(u_i) = 1$ and $\mu_B(u_i)+\nu_B(u_i)+\tau_B(u_i) = 1$, we have that

$$d_s(A,B) = \frac{2}{\pi}\sum_{i=1}^{n} \arccos\left(\sqrt{\mu_A(u_i)\mu_B(u_i)} + \sqrt{\nu_A(u_i)\nu_B(u_i)} + \sqrt{\tau_A(u_i)\tau_B(u_i)}\right)$$

This is summarised in the following definition:

***Definition* 12 (Spherical distance)** *For any two intuitionistic fuzzy sets $A = \{\langle u_i, \mu_A(u_i), \nu_A(u_i)\rangle : u_i \in U\}$ and $B = \{\langle u_i, \mu_B(u_i), \nu_B(u_i)\rangle : u_i \in U\}$ of the universe of discourse $U = \{u_1, u_2, \ldots, u_n\}$, their spherical and normalised spherical distances are:*

- *Spherical distance $d_s(A,B)$*

$$d_s(A,B) = \frac{2}{\pi}\sum_{i=1}^{n} \arccos\left(\sqrt{\mu_A(u_i)\mu_B(u_i)} + \sqrt{\nu_A(u_i)\nu_B(u_i)} + \sqrt{\tau_A(u_i)\tau_B(u_i)}\right)$$

- *Normalised spherical distance $d_{ns}(A,B)$*

$$d_{ns}(A,B) = \frac{2}{n\pi}\sum_{i=1}^{n} \arccos\left(\sqrt{\mu_A(u_i)\mu_B(u_i)} + \sqrt{\nu_A(u_i)\nu_B(u_i)} + \sqrt{\tau_A(u_i)\tau_B(u_i)}\right)$$

Clearly, we have that $0 \leq d_s(A,B) \leq n$ and $0 \leq d_{ns}(A,B) \leq 1$.

Different from the distances in Section 3.3.2, the proposed spherical distances implement in their definition not only the difference between membership, non-membership and hesitancy degrees, but also their actual values. This is shown in the following result:

***Lemma* 10** *$A = \{\langle u_i, \mu_A(u_i), \nu_A(u_i)\rangle : u_i \in U\}$ and $B = \{\langle u_i, \mu_B(u_i), \nu_B(u_i)\rangle : u_i \in U\}$ are two intuitionistic fuzzy subsets of the universe of discourse $U = \{u_1, u_2, \ldots, u_n\}$, and $a = \{a_1, a_2, \ldots, a_n\}$ and $b = \{b_1, b_2, \ldots, b_n\}$ two sets of real numbers (constants). If the following conditions hold for each $u_i \in U$*

$$\mu_B(u_i) = \mu_A(u_i) + a_i$$

$$\nu_B(u_i) = \nu_A(u_i) + b_i$$

then the following inequalities hold

$$\frac{2}{\pi}\sum_{i=1}^{n} \arccos\sqrt{1-e_i^2} \leq d_s(A,B) \leq \frac{2}{\pi}\sum_{i=1}^{n} \arccos\sqrt{(1-c_i)(1-|a_i|-|b_i|)}$$

$$\frac{2}{n\pi}\sum_{i=1}^{n} \arccos\sqrt{1-e_i^2} \leq d_{ns}(A,B) \leq \frac{2}{n\pi}\sum_{i=1}^{n} \arccos\sqrt{(1-c_i)(1-|a_i|-|b_i|)}$$

where, $c_i = \max\{|a_i|, |b_i|\}$ and $e_i = \min\{|a_i|, |b_i|\}$. The maximum distance between A and B is obtained if and only if one of them is a fuzzy set or their available information supports only opposite membership degree for each one.

Proof According to Definition (12), we have

$$d_s(A,B) = \frac{2}{\pi}\sum_{i=1}^{n} \arccos\left(\sqrt{\mu_A(u_i)\mu_B(u_i)} + \sqrt{\nu_A(u_i)\nu_B(u_i)} + \sqrt{\tau_A(u_i)\tau_B(u_i)}\right)$$

Because A and B satisfy

$$\mu_B(u_i) = \mu_A(u_i) + a_i$$

$$\nu_B(u_i) = \nu_A(u_i) + b_i$$

then

$$\tau_B(u_i) = 1 - \mu_A(u_i) - \nu_A(u_i) - a_i - b_i$$

and

$$d_s(A,B) = \frac{2}{\pi}\sum_{i=1}^{n} \arccos\Big(\sqrt{\mu_A(u_i)(\mu_A(u_i)+a_i)} + \sqrt{\nu_A(u_i)(\nu_A(u_i)+b_i)} \\ + \sqrt{(1-\mu_A(u_i)-\nu_A(u_i))(1-\mu_A(u_i)-\nu_A(u_i)-a_i-b_i)}\Big)$$

Let

$$f(u,v) = \sqrt{u(u+a_i)} + \sqrt{v(v+b_i)} + \sqrt{(1-u-v)(1-u-v-a_i-b_i)}$$

Denoting $a = |a_i|$ and $b = |b_i|$, then we distinguish 4 possible cases
Case 1: $a_i \geq 0$ and $b_i \geq 0$. In this case,

$$f(u,v) = \sqrt{u(u+a)} + \sqrt{v(v+b)} + \sqrt{(1-u-v)(1-u-v-a-b)}$$

Let $f'(u,v)|_u = 0$, we have

$$u = \frac{a(v-1)}{b} \text{ and } u = \frac{a(1-a-b-v)}{2a+b}$$

Let $f'(u,v)|_v = 0$, then

$$v = \frac{b(u-1)}{a} \text{ and } v = \frac{b(1-a-b-u)}{2b+a}$$

where

$$u = \frac{a(v-1)}{b} < 0 \text{ and } v = \frac{b(u-1)}{a} < 0$$

hence the valid solutions are

$$u = \frac{a(1-a-b-v)}{2a+b} \text{ and } v = \frac{b(1-a-b-u)}{2b+a}$$

Solving these equations, we have

$$u = \frac{a(1-a-b)}{2(a+b)} \text{ and } v = \frac{b(1-a-b)}{2(a+b)}$$

hence

$$f_0 = f\left(\frac{a(1-a-b)}{2(a+b)}, \frac{b(1-a-b)}{2(a+b)}\right) = \sqrt{1-(a+b)^2}$$

Obviously, the third square root in $f(u,v)$ must be defined, we have

$$0 \leq u \leq 1-a-b \text{ and } 0 \leq v \leq 1-a-b$$

The boundary points are reached when u and v get their minimum or maximum values. Assume $t = \tau_A(u_i)$, they have to satisfy

$$u + v + t = 1$$

if $u = 1 - a - b$ and $v = 1 - a - b$ then $t = 2a + 2b - 1$, we have $(1 - u - v)(1 - u - v - a - b) = (2a + 2b - 1)(a + b - 1) \leq 0$, the third square root in $f(u, v)$ is not defined. therefore, we have three boundary points for A

$$u = 0, \quad v = 1 - a - b, \quad t = a + b$$

$$u = 1 - a - b, \quad v = 0, \quad t = a + b$$

$$u = 0, \quad v = 0, \quad t = 1$$

Thus

$$f_1 = f(0, 1 - a - b) = \sqrt{(1 - a)(1 - a - b)}$$

$$f_2 = f(1 - a - b, 0) = \sqrt{(1 - b)(1 - a - b)}$$

$$f_3 = f(0, 0) = \sqrt{1 - a - b}$$

If $a_i \geq 0$ and $b_i \geq 0$, then $a + b \leq 1$, we have

$$f_1 \leq f_3 \leq f_0$$

$$f_2 \leq f_3 \leq f_0$$

The relationship between f_1 and f_2 depends on the relationship between a and b. Let $c = \max\{a, b\}$ and $e = \min\{a, b\}$, then we have

$$\sqrt{(1 - c)(1 - a - b)} \leq f(u, v) \leq \sqrt{1 - e^2}$$

Case 2: $a_i \leq 0$ and $b_i \leq 0$. In this case, the same conclusion is obtained following a similar reasoning.

Case 3: $a_i \leq 0$ and $b_i \geq 0$. In this case,

$$f(u, v) = \sqrt{u(u - a)} + \sqrt{v(v + b)} + \sqrt{(1 - u - v)(1 - u - v + a - b)}$$

Let $f'(u, v)|_u = 0$, we have

$$u = \frac{a(1 - v)}{b} \text{ and } u = \frac{a(1 + a - b - v)}{2a - b}$$

Let $f'(u,v)|_v = 0$, then

$$v = \frac{b(1-u)}{a} \text{ and } v = \frac{-b(1+a-b-u)}{a-2b}$$

For $u = \frac{a(1-v)}{b}$ and $v = \frac{b(1-u)}{a}$, we get $a = b$, hence

$$u = 1 - v \text{ and } u - a = 1 - v - b$$

Clearly, both A and B are fuzzy sets under this situation. According to lemma 12, we know that

$$\sqrt{1-a} \leq f(u,v) \leq \sqrt{1-a^2}$$

For $u = \frac{a(1+a-b-v)}{2a-b}$ and $v = \frac{-b(1+a-b-u)}{a-2b}$, we have

$$u = \frac{a(1+a-b)}{2(a-b)} \text{ and } v = -\frac{b(1+a-b)}{2(a-b)}$$

if $a > b$

$$u > 0 \text{ and } v < 0$$

otherwise

$$u < 0 \text{ and } v > 0$$

hence the only valid solution exists when $b = 0$, in which case

$$f_0 = \sqrt{1-a^2}$$

For $u = \frac{a(1-v)}{b}$ and $v = \frac{-b(1+a-b-u)}{a-2b}$, we have

$$u = \frac{a(1+b)}{2b} \text{ and } v = \frac{1-b}{2}$$

then

$$f_1 = \begin{cases} \frac{a\sqrt{1-b^2}}{b} & a > b \text{ and } b \neq 0; \\ \sqrt{1-b^2} & a \leq b. \end{cases}$$

For set A, $u + v \leq 1$, and

$$\frac{a(1+b)}{2b} + \frac{1-b}{2} \leq 1$$

then

$$(a-b)(1+b) \leq 0$$

however $1 + b > 0$, which implies $a \leq b$. Therefore we have

$$f_1 = \sqrt{1 - b^2}$$

Similarly, for $u = \frac{a(1+a-b-v)}{2a-b}$ and $v = \frac{b(1-u)}{a}$, we have $a \geq b$ and

$$f_2 = \sqrt{1 - a^2}$$

For u and v under the second situation, we have

$$a \leq u \leq 1 \text{ and } 0 \leq v \leq 1 - \max\{a, b\}$$

therefore, we have boundary points for A

$$u = a, \quad v = 1 - a, \quad t = 0$$

$$u = a, \quad v = 1 - b, \quad t = b - a$$

$$u = 1, \quad v = 0, \quad t = 0$$

$$u = a, \quad v = 0, \quad t = 1 - a$$

Thus

$$f_3 = f(a, 1 - a) = \sqrt{(1 - a)(1 - a + b)}$$

$$f_4 = f(a, 1 - b) = \sqrt{1 - b}$$

$$f_5 = f(1, 0) = \sqrt{1 - a}$$

$$f_6 = f(a, 0) = \sqrt{(1 - a)(1 - b)}$$

Denoting $e = \min\{a, b\}$, we have

$$\sqrt{1 - e^2} \geq f(u, v) \geq \sqrt{\min\{(1 - a)(1 - a + b), (1 - a)(1 - b)\}}$$

Let $c = \max\{a, b\}$ then

$$\sqrt{1 - e^2} \geq f(u, v) \geq \sqrt{\min\{(1 - c)(1 - c + e), (1 - a)(1 - b)\}}$$

Thus

$$\sqrt{1 - e^2} \geq f(u, v) \geq \sqrt{(1 - c)(1 - a - b)}$$

Case 4: $a_i \geq 0$ and $b_i \leq 0$. Again, following a similar reasoning to the one in case 3, the same conclusion can be drawn in this case.

In the four cases we conclude that

$$\frac{2}{\pi}\sum_{i=1}^{n} \arccos\sqrt{1-e_i^2} \leq d_s(A,B) \leq \frac{2}{\pi}\sum_{i=1}^{n} \arccos\sqrt{(1-c_i)(1-|a_i|-|b_i|)}$$

$$\frac{2}{n\pi}\sum_{i=1}^{n} \arccos\sqrt{1-e_i^2} \leq d_s(A,B) \leq \frac{2}{n\pi}\sum_{i=1}^{n} \arccos\sqrt{(1-c_i)(1-|a_i|-|b_i|)}$$

where $c_i = \max\{|a_i|,|b_i|\}$ and $e_i = \min\{|a_i|,|b_i|\}$.

According to the boundary points discussed in the four cases, if $a_i b_i \geq 0$ holds for each u_i, we have

$$\mu_A(u_i) = 0, \quad \nu_A(u_i) = 1-a-b, \quad \tau_A(u_i) = a+b$$

or

$$\mu_A(u_i) = 1-a-b, \quad \nu_A(u_i) = 0, \quad \tau_A(u_i) = a+b$$

hence, the set B has to satisfy

$$\mu_B(u_i) = a, \quad \nu_B(u_i) = 1-a, \quad \tau_B(u_i) = 0$$

or

$$\mu_B(u_i) = 1-b, \quad \nu_B(u_i) = b, \quad \tau_B(u_i) = 0$$

Obviously, B is a fuzzy set. Similar conclusion can be drawn if $a_i b_i < 0$ holds for each u_i and $\sqrt{(1-c)(1-c+e)} \leq \sqrt{(1-a)(1-b)}$. However, if $a_i < 0$ and $b_i > 0$ hold for each u_i and $\sqrt{(1-a)(1-a+b)} > \sqrt{(1-a)(1-b)}$, we have

$$\mu_A(u_i) = a, \quad \nu_A(u_i) = 0, \quad \tau_A(u_i) = 1-a$$

hence, the set B has to satisfy

$$\mu_B(u_i) = 0, \quad \nu_B(u_i) = b, \quad \tau_B(u_i) = 1-b$$

Obviously, A is an intuitionistic fuzzy set with available information supporting membership only, and B is an intuitionistic fuzzy set with available information supporting non-membership only.

Due to the non-linear characteristic of the spherical distance, they do not satisfy lemma 9. However, the following properties hold for the spherical distances:

***Lemma* 11** *$A = \{\langle u_i, \mu_A(u_i), \nu_A(u_i)\rangle : u_i \in U\}$ and $B = \{\langle u_i, \mu_B(u_i), \nu_B(u_i)\rangle : u_i \in U\}$ and $E = \{\langle u_i, \mu_E(u_i), \nu_E(u_i)\rangle : u_i \in U\}$ are three intuitionistic fuzzy* *subsets of the universe of discourse* $U = \{u_1, u_2, \ldots, u_n\}$, *and* $a = \{a_1, a_2, \ldots, a_n\}$ *and* $b = \{b_1, b_2, \ldots, b_n\}$ *two sets of real positive numbers (constants) satisfying the following conditions*

$$|\mu_B(u_i) - \mu_A(u_i)| = a_i, \quad |\nu_B(u_i) - \nu_A(u_i)| = b_i$$

$$|\mu_E(u_i) - \mu_A(u_i)| = a_i, \quad |\nu_E(u_i) - \nu_A(u_i)| = b_i$$

If E *is one of the two extreme crisp sets with either* $\mu_E(u_i) = 1$ *or* $\nu_E(u_i) = 1$ *for all* $u_i \in U$*, then the following inequalities hold*

$$d_s(A, B) < d_s(A, E), \quad d_{ns}(A, B) < d_{ns}(A, E)$$

The distance between intuitionistic fuzzy sets A *and* B *is always lower than the distance between* A *and the extreme crisp sets* E *under the same difference of their memberships and non-memberships.*

Proof We provide the proof just for the extreme fuzzy set with full memberships, the proof for full non-membership being similar.

With E being the extreme crisp set with full memberships, we have

$$\mu_E(u_i) = 1, \quad \nu_E(u_i) = 0, \quad \tau_E(u_i) = 0$$

Because $|\mu_E(u_i) - \mu_A(u_i)| = a_i$ and $|\nu_E(u_i) - \nu_A(u_i)| = b_i$, then we have

$$\mu_A(u_i) = 1 - a_i, \quad \nu_A(u_i) = b_i, \quad \tau_A(u_i) = a_i - b_i$$

From $|\mu_B(u_i) - \mu_A(u_i)| = a_i$ and $|\nu_B(u_i) - \nu_A(u_i)| = b_i$, we have

$$\mu_B(u_i) = 1 - 2a_i, \quad \nu_B(u_i) = 2b_i, \quad \tau_B(u_i) = 2(a_i - b_i)$$

Therefore, we have

$$\begin{aligned} d_s(A, B) &= \frac{2}{\pi} \sum_{i=1}^{n} \arccos\left(\sqrt{(1-a_i)(1-2a_i)} + \sqrt{2b_i^2} + \sqrt{2(a_i - b_i)^2}\right) \\ &= \frac{2}{\pi} \sum_{i=1}^{n} \arccos\left(\sqrt{2}a_i + \sqrt{(1-a_i)(1-2a_i)}\right) \end{aligned}$$

and

$$d_s(A, E) = \frac{2}{\pi} \sum_{i=1}^{n} \arccos\left(\sqrt{1 - a_i}\right)$$

Obviously, we have

$$\sqrt{2}a_i + \sqrt{(1-a_i)(1-2a_i)} > \sqrt{1-a_i}$$

Thus

$$d_s(A,B) < d_s(A,E)$$

and dividing by n

$$d_{ns}(A,B) < d_{ns}(A,E)$$

Lemma 11 shows that the extreme crisp sets with full memberships or full non-memberships are categorically different from other intuitionistic fuzzy sets. With the same difference of memberships and non-memberships, the distance from an extreme crisp set is always greater than the distances from other intuitionistic fuzzy sets. This conclusion agrees with our human perception about the quality change against quantity change, and captures the semantic difference between extreme situation and intermediate situations.

Spherical Distances for Fuzzy Sets

As we have already mentioned, fuzzy sets are particular cases of intuitionistic fuzzy sets. Therefore, the above spherical distances can be applied to fuzzy sets. In the following we provide lemma 12 for the distance between two fuzzy sets.

***Lemma* 12** *Let* $A = \{\langle u_i, \mu_A(u_i)\rangle : u_i \in U\}$ *and* $B = \{\langle u_i, \mu_B(u_i)\rangle : u_i \in U\}$ *be two fuzzy sets in the universe of discourse* $U = \{u_1, u_2, \ldots, u_n\}$*, and* $a = \{a_1, a_2, \ldots, a_n\}$ *is a set of non-negative real constants. If* $|\mu_A(u_i) - \mu_B(u_i)| = a_i$ *holds for each* $u_i \in U$*, then the following inequalities hold*

$$\frac{2}{\pi}\sum_{i=1}^{n} \arccos\sqrt{1-{a_i}^2} \leq d_s(A,B) \leq \frac{2}{\pi}\sum_{i=1}^{n} \arccos\sqrt{1-a_i}$$

$$\frac{2}{n\pi}\sum_{i=1}^{n} \arccos\sqrt{1-{a_i}^2} \leq d_{ns}(A,B) \leq \frac{2}{n\pi}\sum_{i=1}^{n} \arccos\sqrt{1-a_i}$$

The maximum distance between A *and* B *is achieved if and only if one of them is a crisp set.*

Proof According to Definition (12), we have

$$d_s(A,B) = \frac{2}{\pi}\sum_{i=1}^{n} \arccos\left(\sqrt{\mu_A(u_i)\mu_B(u_i)} + \sqrt{\nu_A(u_i)\nu_B(u_i)} + \sqrt{\tau_A(u_i)\tau_B(u_i)}\right)$$

For fuzzy sets, we have

$$\tau_A(u_i) = 0 \text{ and } \nu_A(u_i) = 1 - \mu_A(u_i)$$

$$\tau_B(u_i) = 0 \text{ and } \nu_B(u_i) = 1 - \mu_B(u_i)$$

Considering $|\mu_A(u_i) - \mu_B(u_i)| = a_i$ and $\mu_A(u_i) \geq 0$, we have

$$\mu_B(u_i) = \mu_A(u_i) \pm a_i$$

If

$$\mu_B(u_i) = \mu_A(u_i) + a_i$$

then

$$\begin{aligned} d_s(A,B) &= \frac{2}{\pi}\sum_{i=1}^{n} \arccos\left(\sqrt{\mu_A(u_i)\mu_B(u_i)} + \sqrt{(1-\mu_A(u_i))(1-\mu_B(u_i))}\right) \\ &= \frac{2}{\pi}\sum_{i=1}^{n} \arccos\left(\sqrt{\mu_A(u_i)(\mu_A(u_i)+a_i)} + \sqrt{(1-\mu_A(u_i))(1-\mu_A(u_i)-a_i)}\right) \end{aligned}$$

This can be rewritten as

$$d_s(A,B) = \frac{2}{\pi}\sum_{i=1}^{n} \arccos f_i(\mu_A(u_i))$$

with $f_i(t) = \sqrt{t(t+a_i)} + \sqrt{(1-t)(1-t-a_i)}$, $t \in [0, 1-a_i]$. The extremes of function $f_i(t)$ will be among the solution of

$$f_i'(t) = 0 \;\; t \in (0, 1-a_i)$$

and the values 0 and $1-a_i$, i.e, among $\frac{1-a_i}{2}$, 0 and $1-a_i$. The maximum value $\sqrt{1-{a_i}^2}$ is obtained when $t = \frac{1-a_i}{2}$, while the minimum value $\sqrt{1-a_i}$ is obtained in both 0 and $1-a_i$. We conclude that

$$\frac{2}{\pi}\sum_{i=1}^{n} \arccos\sqrt{1-{a_i}^2} \leq d_s(A,B) \leq \frac{2}{\pi}\sum_{i=1}^{n} \arccos\sqrt{1-a_i}$$

When $t = 0$ or $t = 1-a_i$, we have respectively $\mu_A(u_i) = 0$ and $\mu_B(u_i) = 1-a_i+a_i = 1$, which implies that one set among A and B has to be crisp in order to reach the maximum value under the given difference in their membership degrees.

Following a similar reasoning, it is easy to prove that the same conclusion is obtained in the case $\mu_B(u_i) = \mu_A(u_i) - a_i$. If the last case of bring $\mu_B(u_i) = \mu_A(u_i) - a_i$ for some i, and $\mu_B(u_j) = \mu_A(u_j) + a_j$ for some j, then we could separate the elements into two different groups, each of them satisfying the inequalities, and therefore their summation obviously satisfying it too. The normalised inequality is obtained just by dividing the first one by n.

Because spherical distances are quite different from the traditional distances, the semantics associated to them also differ. For the same relative difference in membership degrees, the spherical distance varies with the locations of its two relevant sets in the membership degree space, 2D for fuzzy sets and 3D for intuitionistic fuzzy sets. The spherical distance achieves its maximum when one of the fuzzy sets is an extreme crisp set. The following example illustrates this effect.

Example Consider our previous example about human behaviour, we can classify our behaviour as perfect, good, acceptable, poor and worst. Their corresponding fuzzy membership as 1, 0.75, 0.5, 0.25 and 0. $A = \{\langle u, 0.75\rangle : u \in U\}$, $B = \{\langle u, 0.5\rangle : u \in U\}$ and $E = \{\langle u, 1\rangle : u \in U\}$ are three fuzzy subsets and $U = \{u\}$ is a universe of discourse with one element only.

From Section 3.3, we have

$$d_1(A,B) = l_1(A,B) = e_1(A,B) = q_1(A,B) = 0.25$$

$$d_1(A,E) = l_1(A,E) = e_1(A,E) = q_1(A,E) = 0.25$$

Obviously, we have

$$d_1(A,B) = d_1(A,E), \quad l_1(A,B) = l_1(A,E)$$

$$e_1(A,B) = e_1(A,E), \quad q_1(A,B) = q_1(A,E)$$

From Definition 12, we have

$$d_{ns}(A,B) = d_s(A,B) = \frac{2}{\pi}\sum_{i=1}^{n} \arccos\left(\sqrt{0.75 * 0.5} + \sqrt{(1-0.75)(1-0.5)}\right) = 0.17$$

$$d_{ns}(A,E) = d_s(A,E) = \frac{2}{\pi}\sum_{i=1}^{n} \arccos\left(\sqrt{0.75 * 1}\right) = 0.33$$

Obviously, the traditional linear distance of fuzzy sets does not differentiate the semantic difference of a crisp set from a fuzzy set. However, $d_s(A,E)$ and $d_{ns}(A,E)$ are much greater than $d_s(A,B)$ and $d_{ns}(A,B)$. It demonstrates that the crisp set E is much more different from A than B although their membership difference appears the same. Hence, the proposed spherical distance does show the semantic difference between a crisp set and a fuzzy set. This is useful when this kind of semantic difference is significant in the consideration.

Figure 4.4 shows four comparisons between the spherical distance and the Hamming distance for two fuzzy subsets A, B with a universe of discourse with one element $U = \{u\}$. The curves represent the spherical distance, and lines denote Hamming distances. Figure 4.4(a) displays how the distance changes with respect to $\mu_B(x)$ when $\mu_A(x) = 0$, fig. 4.4(b) uses the value $\mu_A(x) = 1$, in fig. 4.4(c) the value $\mu_A(x) = 0.2$ is used, and finally $\mu_A(x) = 0.5$ is used in fig. 4.4(d). Clearly, the spherical distance changes sharply for values close to the two lower and upper memberships values, but slightly for values close to the middle membership value. In the case of the Hamming distance, the same rate of change is always obtained.

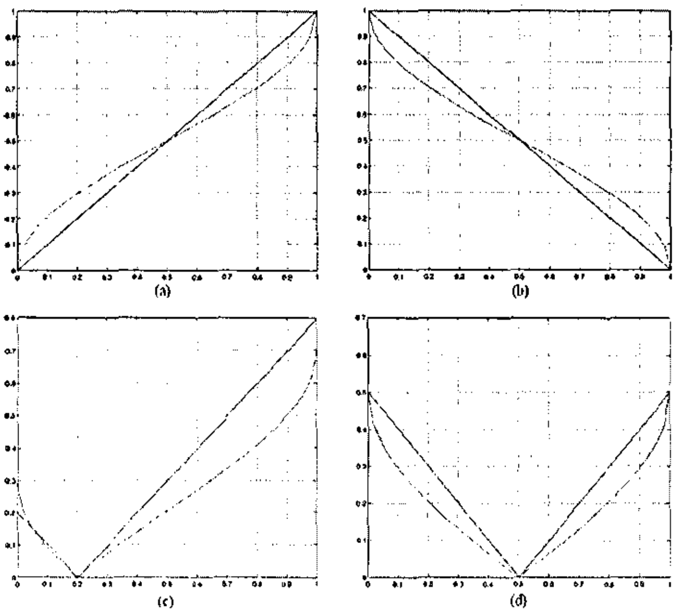
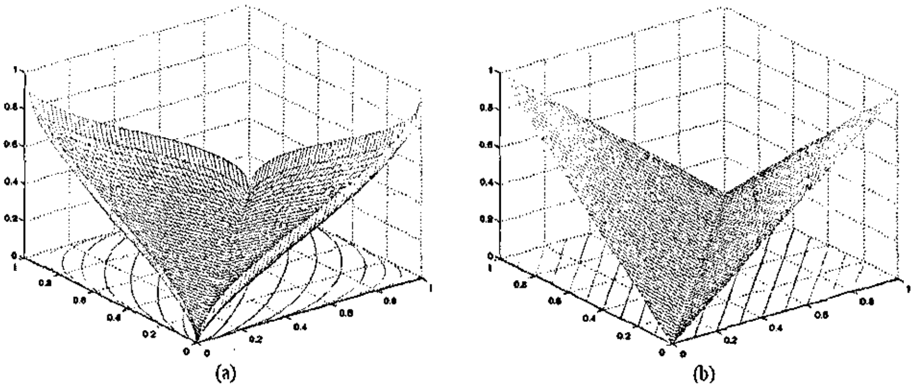


Figure 4.4: Comparison between spherical distances and Hamming distances for fuzzy sets

Figure 4.5 displays how the spherical distance and Hamming distance changes with respect to $\mu_B(x)$ for all possible values of $\mu_A(x)$. Figure. 4.5(a), shows that the spherical distance forms a curve surface, while a plane surface produced by the Hamming distance is shown in fig. 4.5(b). Their contours in the bottom show their differences clearly. The contours for spherical distances are ellipses coming from $(0, 0)$ and $(1, 1)$ with curvatures increasing sharply near $(0, 1)$ and $(1, 0)$. Compared with these ellipses, the contours of Hamming distance are a set of parallel lines. The figures prove our conclusions in lemma 12: the spherical distances do not remain constant as Hamming distances do when both sets experience the same change in their membership degrees.



(a) (b)

Figure 4.5: Grid of spherical distances and Hamming distances for fuzzy sets

4.1.4 Application of distance between intuitionistic fuzzy sets to airport sustainability with respect to aircraft noise

An accurate prediction of aircraft noise has to have accurate input parameters such as engine thrust. In reality, it is very difficult to keep accurate records of thrust for the whole journey, especially during the approach. The thrust is controlled by computers and is changing dynamically according to the wind speed and aircraft direction so as to keep it within the approach zone. Therefore, it is unrealistic to get accurate thrust measurements for the whole journey, and a more realistic requirement would be an interval of thrust rather than an accurate value.

Sustainability refers to the ability of airport to continue its operation without significant degradation of environment in its vicinity. The most notable issue in the environment of an airport is noise disturbance. Therefore, we need to find a way to quantify noise disturbance. Disturbance of aircraft noise at airport is a very complicated issue, it involves not only the measurement of noise at a specific location, but also the different responses of different individuals to the same noise level as well. For the same aircraft, some people may feel it very annoying, but some other people in the same location may not feel it to be a problem at all. Therefore, a one side measurement of noise level only would not be able to reflect the real disturbance of people around the airport in concern. It is necessary to take into account the different perceptions of people to the same noise level.

Given a set of sensitive neighbour areas, the problem is to find a way to map the interaction between noise level and people in these areas. It is clear that fuzzy sets are a good option in setting up this interaction. Each person in the set could have a different membership function, and the same noise level could produce different annoyance, as shown in Figure4.6. Because of the inaccurate thrust, we adopt interval-valued fuzzy sets to consider the inaccurate noise levels caused by the inaccurate thrust. Therefore, the sustainability is converted into a distance to the worst scenario where each person is suffering a full disturbance. This sustainability is a distance to the unsustainable status, so the larger is its value, the more acceptable the airport is with respect to noise disturbance.

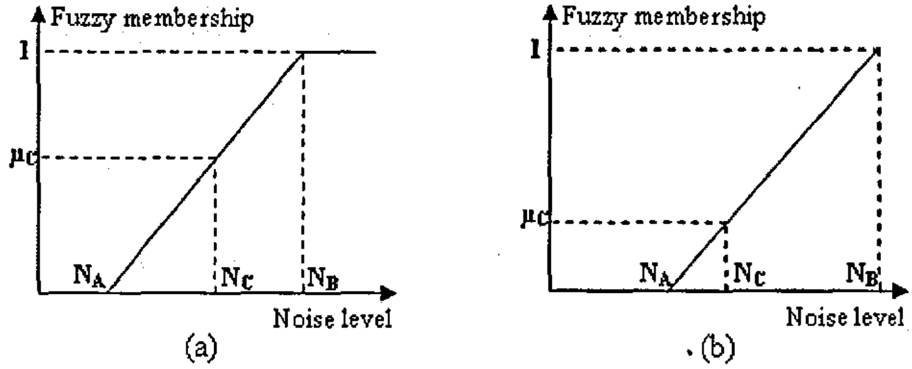


Figure 4.6: Different fuzzy membership values for the same noise level

Based on this idea, the noise disturbed population can be considered as a universe U at an airport. Each person living in this area is an element $x_i \in U$. An interval-fuzzy set $A \subseteq U$ is the set with each element $x \in A$ is associated with an interval as its fuzzy membership $\mu_A(x_i) = [a, b]$, where$0 \leq a \leq b \leq 1$. Considering the interval input of thrust, the output of the neural network for noise level would be an interval as well, hence the membership function of each person shown in Figure 4.6 will be an interval as well. Depending on the understanding of the nature of this interval, it could also be considered as a grey set if the interval represents only a single value. However, the distance between two such sets would follow the same representations due to their same representation form. Here, we do not differentiate the two representations and use the notation of interval-valued fuzzy sets. This set can easily be transformed into intuitionistic fuzzy set and then we can get their distance following equations in Section 4.1.

Consider the worst scenario as set $B \subset U$ where for each element $x_i \in B$, we have

$$\mu_B(x_i) = 1, \quad \nu_B(x_i) = 0, \quad \text{and} \quad \tau_B(x_i) = 0$$

Then we have

$$l_{eh}(A, B) = \frac{1}{n} \sum_{i=1}^{n} max\{|\mu_A(x_i) - 1)|, |\nu_A(x_i)|, |\tau_A(x_i)|\}$$

$$q_{eh}(A,B) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} max\{(\mu_A(x_i)-1)^2, (\nu_A(x_i))^2, (\tau_A(x_i))^2\}}$$

$$d_{ns}(A,B) = \frac{2}{n\pi}\sum_{i=1}^{n} \arccos\left(\sqrt{\mu_A(u_i)}\right)$$

4.2 Significance analysis using neural networks

4.2.1 Relative Strength of Effect

The Kolmogorov mapping neural network existence theorem [5] has proved that: given any continuous function

$$\varphi : I^n \longrightarrow R^m$$

$$Y = \varphi(X)$$

φ can be implemented exactly by a three-layer neural network having n input nodes, 2n+1 hidden nodes and m output nodes. Thus a three layer neural network has the capability to implement any continuous mapping. It is well known that the knowledge representation of ANN is in the form of the connection weights between the nodes of different layers. Hence the relative significance of the individual input nodes for the output value could be identified from the distribution of these connection weights.

Having established a functional input/output relation, which is expressed via the adaptive setting of weights by means of the application of some learning laws within the processing elements of the network, our interest concentrates on searching for a method of identifying what role these different factors play in the total system mechanism. At some stage, after the training process the neural network is no longer allowed to adapt. The output O_k can then be written as (see Fig 4.11)

$$O_k = \frac{1}{1+e^{-u_k}} \tag{4.11}$$

$$\text{where } u_k = \sum_j O_j w_{jk} + \theta_k$$

$$O_j = \frac{1}{1+e^{-u_j}}$$

$$u_j = \sum_i w_{ij} O_i + \theta_j$$

where w is a connected weight, θ is a threshold, O_i is the value of input unit.

Thus we have

$$O_k = \frac{1}{1 + e^{-\sum_j w_{jk} \frac{1}{1+e^{-\sum_i w_{ij} O_i - \theta_j}} - \theta_k}} \tag{4.12}$$

Since the activation function is a Sigmoid function as shown in Equation 4.12, which is differentiable, the variation of O_k with the change of O_i can be calculated by the differentiation of the equation:

$$\begin{aligned} \frac{\partial O_k}{\partial O_i} =& \sum_{j_n} \sum_{j_{n-1}} \cdots \sum_{j_1} W_{j_n k} G(u_k) W_{j_{n-1} j_n} G(u_{j_n}) \\ & W_{j_{n-2} j_{n-1}} G(u_{j_{n-1}}) W_{j_{n-3} j_{n-2}} G(u_{j_{n-2}}) \ldots W_{i j_1} G(u_{j_1}) \end{aligned} \tag{4.13}$$

where $G(u_k) = \frac{e^{-u_k}}{1+e^{-u_k}}^2$, and $O_{j_n}, O_{j_{n-1}}, O_{j_{n-2}}, \ldots, O_{j_1}$ denote the hidden units in the $n, n-1, n-2, \ldots, 1$ hidden layers. Obviously, no matter what the neural network approximates, all items on the Right Hand Side of Equation 4.13 always exist [90]. According to Equation 4.13, a new parameter RSE_{ki} can be defined as the Relative Strength of Effect for input unit i on output unit k.

***Definition* 13 (Relative Strength of Effect (RSE))** *: For a given sample set $S = \{s_1, s_2, s_3, \ldots, s_j, \ldots, s_r\}$, where, $s_j = \{X, Y\}$, $X = \{x_1, x_2, x_3, \cdots, x_p\}$, $Y = \{y_1, y_2, y_3, \cdots, y_q\}$, if there is a neural network trained by BP algorithm with this set of samples, the RSE_{ki} exists as*

$$\begin{aligned} RSE_{ki} =& C \sum_{j_n} \sum_{j_{n-1}} \cdots \sum_{j_1} W_{j_n k} G(u_k) W_{j_{n-1} j_n} G(u_{j_n}) \\ & W_{j_{n-2} j_{n-1}} G(u_{j_{n-1}}) W_{j_{n-3} j_{n-2}} G(u_{j_{n-2}}) \ldots W_{i j_1} G(u_{j_1}) \end{aligned} \tag{4.14}$$

where C is a normalized constant which controls the maximum absolute value of RSE_{ki} as unit, and the function G denotes the differentiation of the activation function. G, W and u are all the same as in Equation 4.13.

It should be noted that the control of RSE is done with respect to the corresponding output unit, which means all RSE values for every input unit on the corresponding output unit are scaled with the same scale coefficient. Hence, it is clear that RSE ranges from -1 to 1.

Compared with Equation 4.13, RSE_{ki} is similar to the derivative except for its scaling value. But it is a different concept from the differentiation of the original mapping function. RSE_{ki} is a kind of parameter which could be used to measure the relative importance of input factors to output units, and it shows only the relative dominance rather than the differentiation of one to one input and output. The larger the absolute value of RSE_{ki} is, the greater the effect the corresponding input unit has on the output unit. The sign of RSE_{ki} indicates the direction of influence, which means positive action applies to the output when $RSE_{ki} > 0$, and negative action when $RSE_{ki} < 0$. Here, positive action denotes that the output increases with increment of the corresponding input, and decreases with reduction of the corresponding input. On the contrary, negative action indicates that the output decreases when the corresponding input increases, and increases when the corresponding input decreases. The output has no relation with the input if $RSE_{ki} = 0$. RSE_{ki} is a dynamic parameter which changes with the variation of input factors. Hence, we can classify the input factors dynamically based on their RSE values, and do the hierarchical analysis with ANN.

According to Equation 4.14, one can calculate the values of RSE_{ki} by the following steps:

1. Enter all the values of the input units, calculate all values of u_j in the hidden units and u_k in the output units by the standard BP method, where u_j represents $u_{j_n}, u_{j_{n-1}}, u_{j_{n-2}}, \ldots, u_{j_1}$

2. Calculate the values of the G function in the output units and hidden units as

$$G(u_k) = \frac{e^{-u_k}}{(1 + e^{-u_k})^2}$$

$$G(u_j) = \frac{e^{-u_j}}{(1 + e^{-u_j})^2}$$

3. Assume RS as a temporary variable in every unit for calculating RSE. For the output units, we have

$$RS(u_k) = G(u_k)$$

4. Calculate the RS values of the units in preceding layers as follows:

$$RS(u_{j_n}) = G(u_{j_n}) w_{j_n k} RS(u_k)$$

5. Calculate the RS values of the units in other hidden layers

$$RS(u_{j_{n-1}}) = G(u_{j_{n-1}}) w_{j_{n-1} j_n} RS(u_{j_n})$$

   Repeat this calculation up to the first hidden layer

6. Calculate the RS_{ki} value as

$$RS_{ki} = W_{ij_1} RS(u_{j_1})$$

7. Assume the number of input units as p, $RS_{kx} = \max\{|RS_{k1}|, |RS_{k2}|, \ldots, |RS_{kp}|\}$

then control the value of RS_{ki} such that

$$RSE_{ki} = \frac{RS_{ki}}{RS_{kx}}$$

In this manner, the RSE_{ki} value can be calculated. Now that the value of RSE_{ki} shows the relative influence, a comparison may be carried out to find the key input units from all input units by their RSE_{ki}. Therefore, the RSE_{ki} may be considered a very important index to evaluate the relative significance of all inputs.

4.2.2 The constant RSE within the concerning scope

The items in the right side of equation 4.14 can be separated as two groups: the first is a group of weights W and the latter is a group related to the differentiation of the Sigmoid function, the activation function (see Figure 4.12).

The weights of a neural network are fixed when the process for learning has been completed, so the values of the first group are fixed when the inputs are changing; whereas, the values of the latter group are varying with the changing of inputs.

Concerning equation 4.14, we can find that the variability of RSE_{ki} is due to the changing of the values in the latter group which vary with the changing of the input values. After a neural network has been trained with samples, the influence of the input on the output should be determined. Because the knowledge provided by the samples is contained in the weights of the neural network, we can find the consequence of the input on the output using these weights. Generally, the scope of the variability of the inputs is considered in small domains in engineering, and we can separate the whole domain into several parts according to the different characteristics of engineering. We can regard the differentiation of the simulated function as constant within one part (piecewise linear) so as to obtain a global relative strength effect which does not vary with the changing of input position in input space.

If we are not concerned with the variability of the influence of the changing of domains of inputs, then we can regard the trained neural network as a linear network, or its activation function is a linear function when we calculate the values of RSEki.

Suppose the activation function as

$$F(x) = x$$

then its differentiation can be obtained as

$$G(x) = 1$$

So, the RSE can be written as

$$RSE_{ki} = C \sum_{j_n} \sum_{j_{n-1}} \cdots \sum_{j_1} W_{j_n k} W_{j_{n-1} j_n} W_{j_{n-2} j_{n-1}} W_{j_{n-3} j_{n-2}} \cdots W_{i j_1} \quad (4.15)$$

The RSE in equation 4.15 stands for the relative importance of every input unit on one output unit in a neural network in the "global" sense, similar to the linear components of the interaction matrix in RES. So, we can define the Global Relative Strength of Effect (GRSE) that the input unit has on the output unit for a certain input domain in a neural network as follows:

***Definition* 14 (Global Relative Strength of Effect (GRSE))** *For a given sample set $S = \{s_1, s_2, s_3, \ldots, s_j, \ldots, s_r\}$, where, $s_j = \{X, Y\}$, $X = \{x_1, x_2, x_3, \ldots, x_p\}$, $Y = \{y_1, y_2, y_3, \ldots, y_q\}$, if there is a neural network trained by BP algorithm with this set of samples, the RSE_{ki} exists as*

$$GRSE_{ki} = C \sum_{j_n} \sum_{j_{n-1}} \cdots \sum_{j_1} W_{j_n k} W_{j_{n-1} j_n} W_{j_{n-2} j_{n-1}} W_{j_{n-3} j_{n-2}} \cdots W_{i j_1} \quad (4.16)$$

where C is a normalized constant which regulates the maximum absolute value of $GRSE_{ki}$ as 1.

The $GRSE_{ki}$ shows the general consequence of every input unit on one output unit in a certain scope of sample space, so it is the preferential parameter for the measuring of importance of input units on output units rather than the actual numerical value of the influence of input units on output units in a specific position in input space. The $GRSE_{ki}$ is a macroscopic or general parameter and the RSE_{ki} is a microcosmic or particular parameter; the former measures the influence of input unit within the entire input space, but the latter does this only at one specific position in input space.

Just like RSE, this GRSE value is also different from the differentiation value. It demands no continuity for the function, so, no matter what the function is, the strength of effect always exists. This is reasonable according to the fact that many functions are not differentiable in some domains or at some points, and the method is suitable for extension to other neural network models.

According to the value of $GRSE_{ki}$, we can assess how much influence the input unit has on the output unit. The more there is revising of the weight due to the input unit , the larger the variance of the weights becomes linked to this input unit. Because the original values of the weights are similar, the larger the absolute values of the weights is, the more the effect of the input unit will have on the output. So, the $GRSE_{ki}$ shows the global dominance of input on output.

The $GRSE_{ki}$ has properties as follows:

- The $GRSE_{ki}$ is a relative value for the comparison of the influence of every input unit versus every output unit;
- The $GRSE_{ki}$ is a value for a certain scope of input space rather than a true value for one specific position in input space;
- The $GRSE_{ki}$ is a measurement of the after effect of one input unit on one of the output units, and the output increases with the increasing of input when $GRSE_{ki} > 0$, and the output decreases with the increasing of input when the $GRSE_{ki} < 0$, and the output has no relation with the input when their $GRSE_{ki} = 0$;
- The absolute value of GRSEki shows how much this input unit controls the output unit compared with other input units, and the larger the absolute value is, the more the influence of this input is.

According to these properties, we can evaluate the importance of system input on system output, or the interaction between inputs and outputs. The larger the absolute value of $GRSE_{ki}$ is, the more important these inputs are. So, we can find out the factors which determine the state of output, and ignore those factors whose $GRSE_{ki}$ is near to zero. Thus, we can control the key factors of engineering so as to make the system behave in the way that we need, or at least to avoid major problems.

4.3 A new method to evaluate a trained artificial neural network

One of the problems in the application of an artificial neural network (ANN) in engineering practice is the difficulty in verifying its function after the training stage. The usual way is to keep some sample data out of its training set so as to test it later. However, it is difficult to decide how much data should be left out since this would reduce the limited data in the training set. On the other hand, it is impossible to test every possible situation in that the artificial neural network is applied usually when we do not know all possible situations. Hence, a simple method which links the general field knowledge and the network structure is better for evaluating the operation of an artificial neural network. Here, a new approach based on the analysis of network structure and field knowledge is presented to help test the function of the artificial neural network.

In comparison with the current local sample testing approach, we propose a new global method to validate the trained artificial neural network. Based on our work on RSE and GRSE, we put forward a new concept - Potential Relative Strength of Effect (PRSE) and the Global PRSE. They provide a link between network structure and field knowledge, which serves as an audit of the trained neural network.

The PRSE and GPRSE are parameters calculated from the network connections distributed in the network, and they reflect the different roles of the input parameters in determining the values of the outputs for the network. From the specific field knowledge or statistics, it is possible to know which parameters are more dominant and important in the process to determine the output. Therefore, the PRSE and GPRSE from a trained artificial neural network should match the specific field analysis if the network is trained successfully. In this way, a complementary method to current testing is provided to give a global view of the behaviour of the trained artificial neural network.

4.3.1 Traditional Validation

The capability to learn from examples by machine without prerequisite knowledge about the specific problem has enabled ANN to become a popular model in engineering applications. Many engineering problems, such as civil engineering, environmental engineering and transportation engineering, involve a number of uncertain mechanisms which complicate the interactions between their different factors. These unknown mechanisms bring the "black box" problems suitable for an ANN to interpret. Because of this kind of incomplete knowledge with respect to the domain problems, the significance of the validation of a trained network appears more important than ever.

There have been numerous different ANN models and a variety of methods for training them; however, the validation of a trained neural network is still carried out using mainly the local sample testing method. This approach randomly separates the entire available data set into two different sets: the training set and the test set. The training set is then used to train the network and the test set is adopted to test the function of the trained network, as shown in Figure 4.7.

In addition to the two sets partition, there is some other validation method which divides the data further, such as the so called cross-validation [87]. This testing method comes from the standard statistics tool cross-validation [159]. After the available data set is randomly partitioned into a training set and a test set, the training set is divided further into two disjoint subsets: an estimation subset and a validation subset. The estimation subset is used to select the

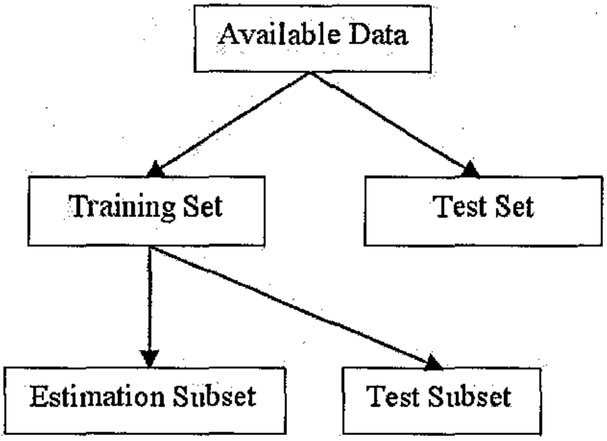


Figure 4.7: The partition of the available data

model (structure and parameters of the network), but the validation subset is used to test or validate the model. The external test set serves for checking the generality of the trained network.

It is obvious that this kind of validation needs a large amount of data. For the available data set, it has to be divided into two or three separate parts and the real data for training the network is only one part of them. However, the data requirement for training a neural network efficiently is also very high. According to the Vapnik-Chervonenkis (VC) dimension [189], the following rule applies [105]:

Let N denote a multi-layer feed forward network whose neurons use a sigmoid activation function

$$\varphi(v) = \frac{1}{1 + e_{-v}}$$

The VC dimension of N is $O(W^2)$, where W is the total number of free parameters in the network. The number of samples needed to learn a mapping reliably is proportional to the VC dimension of that mapping [87], hence the required number of samples for training a neural network is also $O(W^2)$.

In engineering practice, it is sometimes difficult to find sufficient data to train networks. Most data comes from costly measurements carried out on site and involve various uncertainties and complex interactions. With the limited data set, it is difficult to know what the effect would be if only a small part of the available data is used to train the network. There would be two kinds of data set available: ideally distributive data without redundancy and data with repeat and redundancy. For the first group, the pattern existing in the testing data would not be able to be represented in the network trained with the other partial data; and the good agreement for the redundant data in the second group is not reliable for data not considered. The crucial problem for current validation methods is its basic assumption:

- The data in the testing set is representative enough for covering the interesting scope;
- All the patterns in the testing set have been represented in the training set.

In fact, the reason for ANN to be applied is precisely because there is no clear understanding about the mechanism reflected by the data set. Therefore, it is difficult to know if the testing data set has included all possible situations. Considering the potential size of neural networks applied in engineering practice, it is impossible sometimes for the testing set to include all the possible situations within the scope of interest. With limited available data, the more the testing samples are, the less the training data would be and then the poor reliability of the trained network.

Therefore, a better way to check and validate the trained neural network should be to make full use of the available data at the training stage and find the false mapping without or with less involvement of the mass validation data. Here, a new parameter is put forward to find a different approach to evaluate a trained neural network.

4.3.2 PRSE and GPRSE

RSE is a dynamic parameter changing with the variation of the input values of the network. RSE is sensitive to a change in sign of the connecting weight and output of each individual node. However, the absolute value of the connecting weight and individual node is more meaningful in the wider context, indicating the potential maximum capability for a relevant factor to control the corresponding output.

Hence a new parameter can be defined which can judge the degree of importance of a variable on the system, measuring the relative significance of inputs with respect to outputs in the trained neural network. On the basis of RSE, the Potential RSE and Global Potential RSE (GPRSE) have been defined.

***Definition* 15 (Potential RSE (PRSE))** *: For a neural network trained using the BP algorithm and for a given reference data set* $S = \{s_1, s_2, s_3, \ldots, s_j, \ldots, s_r\}$, *where,* $s_j = \{X, Y\}$, $X = \{x_1, x_2, x_3, \ldots, x_p\}$, $Y = \{y_1, y_2, y_3, \ldots, y_q\}$*:*

$$PRSE_{ki} = \frac{\sum_{j_n} \sum_{j_{n-1}} \cdots \sum_{j_1} |W_{j_n k}||G(u_k)||W_{j_{n-1} j_n}||G(u_{j_n})| \ldots |W_{i j_1}||G(u_{j_1})|}{\sum_I \sum_{j_n} \sum_{j_{n-1}} \cdots \sum_{j_1} |W_{j_n k}||G(u_k)||W_{j_{n-1} j_n}||G(u_{j_n})| \ldots |W_{I j_1}||G(u_{j_1})|} \tag{4.17}$$

where, $I = 1, 2, 3, \ldots, p$, and the function G denotes differentiation of the activation function. W is a connected weight and e is the input value in its corresponding node.

***Definition* 16 (Global Potential RSE (GPRSE))** *: For a neural network trained using the BP algorithm and for a given reference data set $S = \{s_1, s_2, s_3, \ldots, s_j, \ldots, s_r\}$, where, $s_j = \{X, Y\}$, $X = \{x_1, x_2, x_3, \ldots, x_p\}$, $Y = \{y_1, y_2, y_3, \ldots, y_q\}$:*

$$GPRSE_{ki} = \frac{\sum_{j_n} \sum_{j_{n-1}} \cdots \sum_{j_1} |W_{j_n k}||W_{j_{n-1} j_n}| \cdots |W_{i j_1}|}{\sum_I \sum_{j_n} \sum_{j_{n-1}} \cdots \sum_{j_1} |W_{j_n k}||W_{j_{n-1} j_n}| \cdots |W_{I j_1}|} \tag{4.18}$$

where, W is a connected weight and e is the input value in its corresponding node.

PRSE and GPRSE are measures of the absolute value of every weight and node value. The absolute influence of every connection and node is thus accumulated. Hence, no matter which factors are dominant, the contribution of every factor will be incorporated within the calculation of PRSE and GPRSE.

Compared with RSE, the removal of the different signs makes the PRSE and GPRSE less sensitive to a small change of input, thus they are measures of the potential within a wider scope of neighbourhood rather than a detailed trend at a specific point.

4.3.3 Validation with PRSE

In engineering practice, the exact independence between different factors represented by RSE is difficult to know because of our ignorance of the complicated interactions. However, the statistics and expertise often have the ability to know roughly which factor is dominant and their relative importance index. In this sense, it is possible to know the GPRSE for the interest scope even before we begin to train the neural networks. This provides us with an alternative for evaluating and validating a trained neural network.

As we know, a suitably trained neural network is able to map the relationships between its input factors and output attributes. This efficient mapping has a precondition: the network can recognise the different roles of the different factors for its mapping function. An important factor should be able to play a significant role when a suitable input is fed into the network. The GPRSE should agree with the field knowledge obtained from statistics or expertise. Hence, a comparison between the GPRSE and the field knowledge about the dominance of different factors in the system would help us to evaluate the trained network.

The GPRSE is defined as a global parameter within the scope of interest, and is capable to indicate the general significance of the individual factors.

However, some relationships may be very complicated and a general validation is not sufficient to audit network behaviour, and the PRSE for some special points would be helpful to validate its function in some special segments. As we know from Chapter 3, the PRSE relies on the specific point in input space but is not so sensitive to its position changes like RSE. It reflects a potential dominance within a wider scope of the neighbourhood of the input point. The scheme to evaluate and validate a trained neural network with GPRSE and PRSE is shown in Figure 4.8.

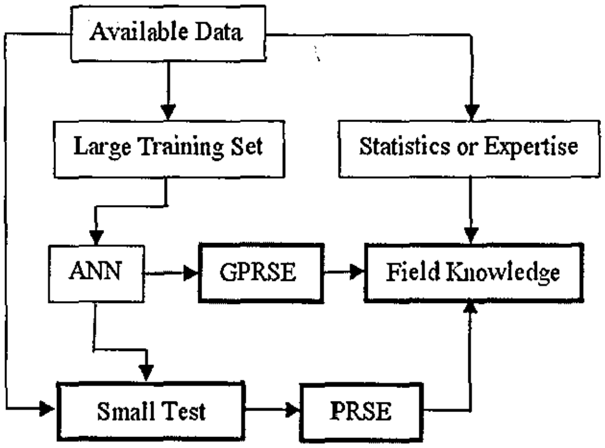


Figure 4.8: The evaluation scheme

Similar to the traditional test, the available data are separated into two different sets: the training set and the test set. However, the number of the samples in the test set could be reduced dramatically if suitable field knowledge about the specific problem is possible. Although satisfying the training data, the false mapping would produce some false details in the resultant mapping relationships. There would be some distortions of the connection weights to provide such kinds of false detail. These changed connections would change their absolute significance and thus be reflected in the changes of GPRSE. Therefore, the disagreement between the GPRSE from the trained network and the field knowledge from statistics and expertise indicates the unreliable mapping of the trained network.

The trends audit with GPRSE would prevent most of the false mappings. Only the false mapping that possesses similar GPRSE values with our knowledge could survive this evaluation. The PRSE audit could be conducted with some key samples where field knowledge is possible. Compared with RSE, PRSE reflects the potential dominance of the factor within a wider scope and hence provides an indicator about local distortions. Those false mappings satisfying the GPRSE would be exposed if the specific field knowledge is available in a few local segments. In this way, the possible false mappings could be filtered dramatically without using large amounts of testing data. Hence, most data could be adopted as training data and the feasibility of the trained network could be improved dramatically. It should be noted that the small test data set for PRSE needs only the relevant dominance rather than the corresponding output. However, it can also serve for traditional testing as a supplement to the proposed method if their outputs are known.

In this way, the available data set is fully applied to the training stage and hence improves the reliability of the trained network considerably. The evaluation and validation of the trained network is carried out mainly with the global parameters GPRSE, as well as the PRSE for some key points.

4.4 The role of redundant structure of neural networks

One of the difficulties in establishing a Neural Network (NN) is the determination of its structure. It is a common understanding that only the simplest network structure can give the best solution. Therefore, various network pruning technologies have been developed [129, 146, 85, 83, 118, 116, 155]. However, one of the key features in neural networks is that they perform complicated analyses or mapping by means of a combination of huge amounts of simple neurons [90]. The real biological world does not necessarily rely on strict mathematics or pruning technology to run their activities, but they do display such an array of perfect functions that scientific method may never be able to explain adequately. The 'compound eye' of an insect [101] is just one of these amazing facts: in addition to its ability to accommodate overlapping inputs, it involves many other different mechanisms which makes it impossible to simulate with only a simple structure. This fact does not exclude the notion that simple overlapped inputs may contribute to its powerful function.

Bearing this in mind, a simple approach to making use of overlapped or redundant inputs to improve the training results of NN is put forward here. This method employs multiple input nodes for the same input parameter in the network structure and simulates their influences in the compound eye of insects [201] by a random initialisation of their connecting weights. Unlike redundant hidden nodes, overlapped inputs do not produce more dimensions in the solution space and hence do not involve new uncertainties. In principle, we prove that the proposed overlapped input structure could be replaced exactly by an equivalent ordinary neural network structure. However, the difficulty in initialising the connecting weights between different nodes makes it very rare for an ordinary network to find the same ideal solution. With the increment of the number of the overlapped input nodes in the input layer, the distribution of the ideal connecting weights tends to approach zero, and hence a random initialisation around zero would appear to be suitable for the proposed simple structure. This conclusion does not exist in traditional neural networks. Therefore, the proposed method has more chance to find the ideal solution with the same available training data than the traditional one. Because of the different initialisation of the connecting weights, the same input from different input nodes may have different influences to the training operation. This difference may reduce with the training process, but it would not disappear completely. Therefore, the node for the same input may "see" different "pictures" and reflects the position effects of the compound eye of insects in some senses.

In conclusion, a simple partition example is illustrated to show the efficiency of the proposed method. The details neglected by traditional networks are revealed clearly using the proposed networks. This shows that a simple overlapped input does improve the training of the neural networks.

4.4.1 Compound eye and redundant structure

It is well known that insects are very sensitive to objects moving around them. Research shows that Drosophila, the fruit fly, has a reiterated pattern of 800 ommatidia in its compound eye [101], and the lacewing Mallada basalis (Walker) has approximately 600 ommatidia [201]. There is a lens in every ommatidium and hence the compound eye is composed of a large number of lenses. Instead of one lens they see through spheres with many lenses. Each lens of the compound eye catches its own image. The more lenses the compound eye bears, the higher the resolution of the image. The two large spherical eyes of a fly give an almost complete 360 degree vision.

The mechanisms of the compound eye are very complicated and still being analysed although some have been recognised. For example, a well-focused clear zone diurnal eye of the Skipper butterfly is illustrated in Figure 4.9 [93]. The parallel rays falling on the eye pass through many facets to converge on a small region of the receptor layer.

The individual ommatidium of the insect's compound eye possesses only a few photoreceptors. For example, there are 8 photoreceptors in the adult Drosophila ommatidium [197]. Obviously, the single ommatidium cannot catch very much information about its view. However, a large number of them makes the insect very sensitive to its visual environment. It proves that the combination of the large number of ommatidia improves dramatically the function of the

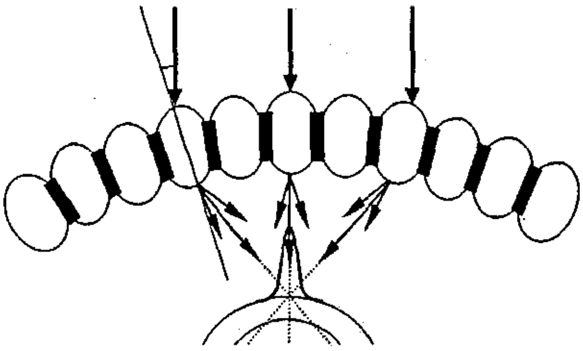


Figure 4.9: A well-focused clear zone eye

insect eye. There are some differences between the images captured by different ommatidia from different angles; however, the main image taken by these ommatidia are similar to each other in that they come from the same picture. In another words, there is some kind of redundant structure in the compound eye of insects, and that structure contributes to the combination of the final image.

Inspired by the compound eye of insects, we constitute a redundant structure for neural networks, as shown in Figure 4.10. Figure 4.10 illustrates the structure of a NN with redundant inputs and an ordinary NN. A, B and C are the attributes of the observed object, and they serve as the inputs of the neural network. For the ordinary one, there are only three input nodes in this case: A, B and C. There are 9 nodes in the input layer of the redundant one for the same object here, and it corresponds to three lenses in the compound eye. The first three input nodes act as one "lens", the second three nodes as another lens, and so do the other three nodes in the input layer of the NN with redundant inputs. In this way, the redundant NN could "see" three similar "images" for the same sample data at the same time, just like the insect's mosaic image from its compound eye. In this way, the input "image" is multiplied as a number of similar "images" from a series of input "lenses", and their messages are projected to the hidden layer to combine into a single "picture". Due to the random initialisation of the connecting weights, it is acknowledged that the different "lens" would "see" a different "image". This difference could be reduced with the learning operation, but it is difficult to remove completely statistically. Thus to some extent this mechanism simulates the operation of the compound eye of an insect.

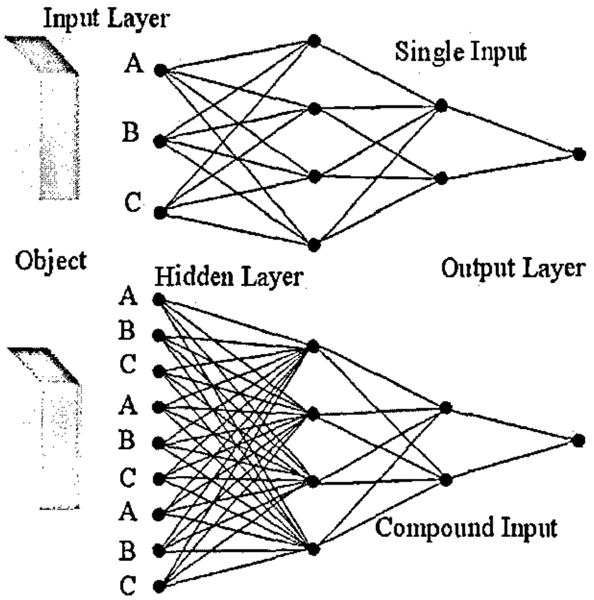


Figure 4.10: Redundant input NN and ordinary NN

4.4.2 Properties of the redundant structure

According to Figure 4.10, it is obvious that the structure of the hidden layers and the output layer of the two different kinds of structure are exactly the same structure: they have the same number of nodes and connecting weights. The only difference comes from the input layer and their connecting weights with the first hidden layer. In principle, there should be an equivalent ordinary network structure to the redundant one.

Considering a network with n input factors, suppose there are k input sets in its corresponding redundant network, W_{ij} is the connecting weight between input factor i and node j in the first hidden layer, Vi denotes the value in input node i, then the receipt of the hidden node j in the redundant network is:

$$V_{ij} = \sum_{i=1}^{n\times k} w_{ij} V_i + \sigma_i$$

Change the order of the input factors so that all the first n input nodes are from factor 1, and the second n input nodes from factor 2, and so on. Then we have Equation 4.19.

$$\begin{aligned} V_{ij} &= \sum_{i=1}^{n\times k} w_{ij} V_i + \sigma_j \\ &= v_1 \sum_{i=1}^{k} w_{ij} + v_2 \sum_{i=k+1}^{2} k w_{ij} + \ldots + v_k \sum_{i=(n-1)\times k+1}^{n\times k} w_{ij} + \sigma_j \qquad (4.19) \\ &= \sum_{s=1}^{n} w'_{sj} v_s + \sigma_j \end{aligned}$$

Here, w_{ij} means the connection between input node i and the hidden node j after the changing of the order for input nodes. represents the value of input factor s. w'_{sj} is the connecting weight between input s and hidden node j in the equivalent ordinary network.

$$w'_{sj} = \sum_{(s-1)\times k+1}^{s\times k} w_{ij} \qquad (4.20)$$

According to Equations 4.19 and 4.20, the compound input ANN can be converted exactly into an equivalent ordinary one. Therefore, it does not introduce any new parameters, thus does not bring new uncertainties to the network. This is very different from a pure increment of the hidden nodes where new uncertainties are inevitable.

Suppose that a global minimum point in the error space requires the connecting weight between input factor m and node n in the first hidden layer to be W'_{mn} in the ordinary NN, and w'_{in} (i=1, 2, ,k) for the compound input, where k means the number of the sub networks. Then

$$W'_{mn} = \sum_{i=1}^{k} w'_{in}$$

Suppose the initialised value of W'_{mn} is W_{mn}, and w_{in} for w'_{in}. Considering the random feature of W_{mn} and w_{in}, thus

$$E(w_{in}) = E(W_{mn}) = 0$$

and

$$\lim_{k\to\infty} \sum_{i=1}^{n} w_{in} = 0$$

Therefore, the increment of the number of nodes for the same input factor does not increase the initial weight of its corresponding traditional NN.

According to the Jaynes' Maximum Entropy Principle [100], we should assume that $w_{in} = \frac{W_{mn}}{k}$ when we do not know the value of every w_{in}. It is clear that

$$\lim_{k \to \infty} w_{in} = \lim_{k \to \infty} \frac{W_{mn}}{k} = 0$$

Therefore, a large number of sub networks in the redundant input NN would move the start point towards the true global minimum in the error space under the condition of a small value initialisation within [-1,1]. The true global minimum in the error space is a precondition for NN to give a reliable solution to its mapping. The random initialisation of weights within the neighbourhood of 0 is more reasonable for a redundant input NN than an ordinary one.

4.5 Data mining using neural networks

As a candidate for data mining in rich data situations, neural networks (NN) have received great attention in recent years. Various models for extracting rules or knowledge from a trained neural network are presented. Andrew et al. provides a very good review of these methods [10]. Most of these existing methodologies focus on the transformation from implicit neural network knowledge representation to explicit rule based structures [183, 11, 62, 63, 115, 54, 157, 136, 156]. This is preferable but difficult to realise for a complicated domain where a solution surface may be approached only by a very fine granularity of space division. Although NN has provided us with a powerful capability to map the input-output relationship, what it does not do well is its self explanation. It is its lack of explanation mechanism that prohibits many potential applications [181].

Here we will not attempt to extract all the knowledge implicitly represented by NN. Instead, we will try to provide some understandable explanation to the mapping results of a trained NN on the basis of its weight connections and training samples.

4.5.1 Dynamic state space

In traditional Euclidean space, all dimensions have equivalent significance. Therefore, coordinates x, y and z make exactly the same contribution in determining the position of a point in 3D space. However, the significance of different input factors are different in neural networks, hence their contribution in determining output are not equivalent. Here, we define a new space for searching the relevant points.

For a given problem domain Ω, there are a number of indicators to describe the features of a case C_j $(0 < j < 8)$ in Ω: $i_1, i_2, \ldots, i_n$, n is the number of features necessary to discriminate C_j from C_k $(k \neq j)$. A Dynamic State

Space (DSS) is a space defined in Ω with $i_1, i_2, \ldots, i_n$ as its dimensions. The significance of dimensions is $s_1, s_2, \ldots, s_n$. For a point in DSS, its necessary number of dimensions n and significance s_l $(0 < l < n)$ is subject to change with time and relevant problem domain.

DSS is clearly different from traditional Euclidean space in that the significances of its dimensions are changeable and its dimension number n depends on the problem's domain.

The distance between two points is defined as:

$$D = \sqrt{|S_1|(i_{1_a} - i_{1_b})^2 + |S_2|(i_{2_a} - i_{2_b})^2 + \ldots + |S_n|(i_{n_a} - i_{n_b})^2}$$

where the subscripts a and b denote two different points in DSS.

With DSS, the training and test samples in NN could be considered as a set of points in DSS with their inputs and outputs as coordinates of corresponding dimensions. Therefore, a set of samples in neural network training and testing would become a set of points in DSS. The significance of dimensions in DSS are different from those in traditional space and the necessary dimensions in searching some relevant points also depend on the searching condition. The points and dimensions with different significance are shown in Figure 4.11.

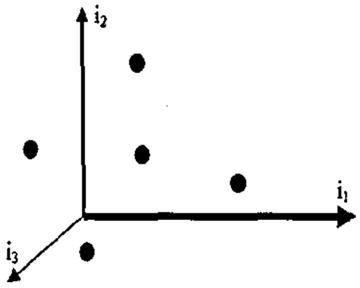


Figure 4.11: Points in dynamic state space

A search in DSS for a desired set of points could be conducted in two different ways:

- Dimension path: search only one dimension every time and stop when there is only one point left or no further dimensions are available. The search order follows the significance of the dimensions, the most significant dimension would be called first.

- Distance path: calculate the distance in DSS for all available points and order the result accordingly. A predefined threshold is adopted to cut those irrelevant points.

In this way, the relevant points in the available data set could be located quickly and provide the basis for explaining a new case. The result for a new case must come from its relevant case, or the relevant points in DSS. The significance of the dimensions indicates the possible change of a new case compared to its relevant old case, thus providing a mechanism to explain the mapping result of neural networks.

4.5.2 Query based on DSS

Having obtained the mapping results and RSE from a trained NN, the dominances of the different attributes are known. Due to the possible complexity of real world problems mapped by NN, it is difficult and unnecessary to trace all the possible rules hidden in the connecting weights. As an intuitive mapping tool, NN has the capability to give an acceptable approximation of the mapping operation. Our task here is to provide a reasonable and understandable explanation of the intuitive reasoning of NN.

Suppose an active case C is fed into the network so as to obtain a mapping output OC. We need to find a reasonable explanation for the output OC. The operational process for extracting this explanation from a trained neural network is illustrated in Figure 4.12.

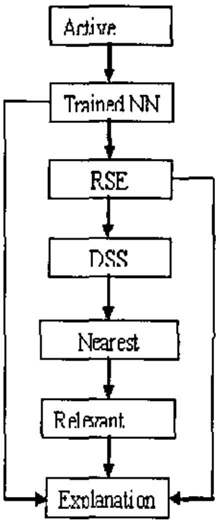


Figure 4.12: The DSS explanation

The active case C is fed first into the trained neural network and its RSE is then adopted as significance of dimensions in DSS. Then a search for relevant cases in the training sample is carried out in DSS with the dimension path or distance path. The result of this search would be a set of cases which is similar to the new case. A comparison is conducted between the output of the neural network for the new case and the similar cases found with the DSS search. There are two possible results for this comparison:

- One similar case is found from the training cases. Its output attribute may or may not be similar to the new case.
- Several similar cases are found, and they may or may not share the similar output with the new case.

For those relevant cases with similar output as the new case, they are a very good explanation of the result: this is based on the existing cases, and they share the similar output because they have similar inputs, most notably sharing similar inputs for those significant attributes. If their outputs are different, then RSE analysis is called in to explain the difference: the attribute with a bigger RSE and output difference would be considered first. Sometimes, however, the combination of small RSE attributes may change the results, hence combined RSE may be taken into account when RSE itself cannot explain the result. When more than one relevant case is left, the explanation would prefer those with a similar output like the new case. If none of them agrees with the new case, a similar RSE analysis would be called in to explain the result.

4.6 Roughness bounds of rough sets

The real world is inherently uncertain, imprecise and vague. Rough set theory focuses on the uncertainty caused by indiscernible elements with different values in decision attributes. It approximates the underlying set with two crisp sets. Therefore, the cardinality of elements in these two sets has a direct influence on the uncertainty of their corresponding rough set as a whole. Consequently, it is important that we consider the roughness of a rough set to have some understanding of the results in any decision making system. Knowing some bounds of this roughness before implementing the set operations can be important. Much research has been carried out on rough set theory, applications and their combination with fuzzy sets [51, 52, 53, 96, 97, 106, 130, 131, 143, 152, 196, 199, 230, 231]. As for roughness, there has been some research on the roughness of fuzzy sets [19, 24, 233]. However, as an important feature of rough sets, roughness has not yet received sufficient attention. Our research supplements this research by investigating the bounds of roughness for rough sets.

Because elements of rough sets are essentially uncertain we have to consider different set operations (e.g. Union, Intersection, Difference and Complement). This section considers the roughness of these operations.

4.6.1 Union of rough sets

For the roughness of approximation on union of rough sets, we have the following theorem.

Theorem 1 *The following inequality holds for the roughness of approximation for a union of two rough sets* $X = X_1 \cup X_2$

$$R_A^\circ(X_1) + R_A^\circ(X_2) - 1 \le R_A^\circ(X) \le \frac{1 - R_A^\circ(X_1)R_A^\circ(X_2)}{2 - R_A^\circ(X_1) - R_A^\circ(X_2)}$$

Proof The roughness of approximation of the union set can be expressed as

$$\begin{aligned} R^\circ{}_A(X) &= \frac{|A^*(X_1) \cup A^*(X_2)| - |A_*(X_1) \cup A_*(X_2)|}{|A^*(X_1) \cup A^*(X_2)|} \\ &= 1 - \frac{|A_*(X_1) \cup A_*(X_2)|}{|A^*(X_1) \cup A^*(X_2)|} \end{aligned}$$

Cardinality represents the number of elements in a set. For any two crisp sets A and B, we have $|A \cup B| \le |A| + |B|$ and $|A \cup B| \ge \max\{|A|, |B|\}$. Therefore

$$R_A^\circ(X) \le 1 - \frac{\max\{|A_*(X_1)|, |A_*(X_2)|\}}{|A^*(X_1)| + |A^*(X_2)|}$$

In order not to lose the generality, assume that $|A_*(X_1)| \ge |A_*(X_2)|$, then

$$\begin{aligned} R_A^\circ(X) &\le 1 - \frac{|A_*(X_1)|}{|A^*(X_1)| + |A^*(X_2)|} \\ &\le 1 - \frac{1}{\frac{|A^*(X_1)|}{|A_*(X_1)|} + \frac{|A^*(X_2)|}{|A_*(X_1)|}} \end{aligned}$$

Because that $|A_*(X_1)| \ge |A_*(X_2)|$, then $\frac{|A^*(X_2)|}{|A_*(X_1)|} \le \frac{|A^*(X_2)|}{|A_*(X_2)|}$. Considering that $\alpha_A(X_1) = \frac{|A_*(X_1)|}{|A^*(X_1)|}$ and $\alpha_A(X_2) = \frac{|A_*(X_2)|}{|A^*(X_2)|}$, we have

$$R_A^\circ(X) \le 1 - \frac{\alpha_A(X_1)\alpha_A(X_2)}{\alpha_A(X_1) + \alpha_A(X_2)}$$

Here, $\alpha_A(X_1)$ and $\alpha_A(X_2)$ mean the accuracy of the approximation for X_1 and X_2. Considering their relationship with the roughness of approximation $\alpha_A(X) = 1 - R_A^\circ(X)$, we have

$$R_A^\circ(X) \le \frac{1 - R_A^\circ(X_1)R_A^\circ(X_2)}{2 - R_A^\circ(X_1) - R_A^\circ(X_2)}$$

The same conclusion will be drawn if we assume $|A_*(X_2)| > |A_*(X_1)|$. In a similar way, we can prove the lower limit

$$\begin{aligned} R_A^\circ(X) &= 1 - \frac{|A_*(X_1)| \cup |A_*(X_2)|}{|A^*(X_1)| \cup |A^*(X_2)|} \\ &\geq 1 - \frac{|A_*(X_1)| + |A_*(X_2)|}{\max\{|A^*(X_1)|, |A^*(X_2)|\}} \end{aligned}$$

Assume $|A^*(X_1)| > |A^*(X_2)|$, then

$$\begin{aligned} R_A^\circ(X) &\geq 1 - \frac{|A_*(X_1)| + |A_*(X_2)|}{|A^*(X_1)|} \\ &\geq R_A^\circ(X_1) + R_A^\circ(X_2) - 1 \end{aligned}$$

Similar to the upper limit, the assumption for $|A^*(X_1)| > |A^*(X_2)|$ does not influence the final result.

This theorem can also be applied to accuracy of the approximation.

***Corollary* 2** *The accuracy of approximation for a union of two rough sets satisfies*

$$\alpha_A(X_1) + \alpha_A(X_2) \geq \alpha_A(X) \geq \frac{\alpha_A(X_1)\alpha_A(X_2)}{\alpha_A(X_1) + \alpha_A(X_2)}$$

We can also extend this theorem to a union of more than two rough sets:

***Lemma* 13** $X_1 \in U, X_2 \in U, \ldots, X_n \in U$ *are* n *sets, and their roughness of approximation are* $R_A^\circ(X_1), R_A^\circ(X_2), \ldots, R_A^\circ(X_n)$. *Their union set* X *has a roughness of approximation* $R_A^\circ(X)$, *then* $R_A^\circ(X)$ *satisfies the following condition:*

$$R_1 \leq R_A^\circ(X) \leq R_2$$

where

$$R_1 = \sum_{i=1}^{n} R_A^\circ(X_i) - n + 1$$

and

$$R_2 = 1 - \frac{\prod_{l=1}^{n}(1 - R_A^\circ(X_l))}{\sum_{i=1}^{n}[\prod_{j=1}^{i-1}(1 - R_A^\circ(X_j))(\prod_{j=i+1}^{n}(1 - R_A^\circ(X_j))]}$$

The proof for this lemma is similar to Theorem 1 and this lemma can also be applied to accuracy of approximation.

***Corollary* 3** $X_1 \in U, X_2 \in U, \ldots, X_n \in U$ *are* n *sets, and their accuracy of approximation are* $\alpha_A(X_1), \alpha_A(X_2), \ldots, \alpha_A(X_n)$. *Their union set* X *has an accuracy of approximation* $\alpha_A(X)$, *then* $\alpha_A(X)$ *satisfies the following condition:*

$$\sum_{i=1}^{n} \alpha_A(X_i) \geq \alpha_A(X) \geq \frac{\prod_{i=1}^{n} \alpha_A(X_i)}{\sum_{i=1}^{n}[\prod_{j=1}^{i-1} \alpha_A(X_j) \prod_{j=i+1}^{n} \alpha_A(X_j)]}$$

4.6.2 Intersection of rough sets

For the intersection of two rough sets, the roughness for the resultant rough set is determined not only by the roughness of the two operand sets, but also the distribution of the two lower approximation of the two operand sets. Indeed, the roughness of approximation for the intersection set can be 0 or 1 regardless of the roughness values of two rough sets. As an example, we consider the intersection set X between rough sets X_1 and X_2.

$$\begin{aligned} R^\circ_A(X) &= \frac{|A^*(X_1) \cap A^*(X_2)| - |A_*(X_1) \cap A_*(X_2)|}{|A^*(X_1) \cap A^*(X_2)|} \\ &= 1 - \frac{|A_*(X_1) \cap A_*(X_2)|}{|A^*(X_1) \cap A^*(X_2)|} \end{aligned}$$

According to the definition of rough sets, $A^*(X_1) \cap A^*(X_2) \supseteq A_*(X_1) \cap A_*(X_2)$. We have $|A^*(X_1) \cap A^*(X_2)| \geq |A_*(X_1) \cap A_*(X_2)|$. Thus $0 \leq \frac{|A_*(X_1) \cap A_*(X_2)|}{|A^*(X_1) \cap A^*(X_2)|} \leq 1$. No matter what roughness values for the two operand sets, we have

$$\frac{|A_*(X_1) \cap A_*(X_2)|}{|A^*(X_1) \cap A^*(X_2)|} = 0, \quad \text{if } |A_*(X_1) \cap A_*(X_2)| = 0$$

$$\frac{|A_*(X_1) \cap A_*(X_2)|}{|A^*(X_1) \cap A^*(X_2)|} = 1, \quad \text{if } |A_*(X_1) \cap A_*(X_2)| = |A^*(X_1) \cap A^*(X_2)|$$

Consequently, we have $0 \leq R^\circ_A(X) \leq 1$.

This particular case illustrates that in general the roughnesses of the two rough sets can not bound the roughness of their intersection. Obviously, the same conclusion holds for the accuracy of approximation and the roughness of the intersection of more than two rough sets.

4.6.3 Difference of two rough sets

The upper and lower limits for difference set is provided by the following theorem.

Theorem 2 *The roughness for approximation of the difference set of two rough sets satisfies the following rules*

- *If* $|A^*(X_2)| \leq |A_*(X_1)|$, *then*

$$\frac{R^\circ_A(X_1) + (R^\circ_A(X_1) - 1)s_*}{1 + (R^\circ_A(X_1) - 1)s_*} \leq R^\circ_A(X) \leq R^\circ_A(X_1) + s^*$$

- *If* $|A^*(X_2) \geq |A_*(X_1)|$ *and* $|A_*(X_2)| \leq |A^*(X_1)|$*, then*

$$\frac{R^\circ_A(X_1) + (R^\circ_A(X_1) - 1)s_*}{1 + (R^\circ_A(X_1) - 1)s_*} \leq R^\circ_A(X) \leq 1$$

- *if* $|A^*(X_2)| \geq |A_*(X_1)|$ *and* $|A_*(X_2)| \geq |A^*(X_1)|$*, then the roughness of approximation of operands can not bound the roughness of approximation for the difference set under this situation*

where, $s_* = \frac{|A_*(X_2)|}{|A_*(X_1)|}$ *and* $s^* = \frac{|A^*(X_2)|}{|A^*(X_1)|}$.

Proof

$$\begin{aligned} R^\circ &= \frac{|A^*(X_1) - A_*(X_2)| - |A_*(X_1) - A^*(X_2)|}{|A^*(X_1) - A_*(X_2)|} \\ &= 1 - \frac{|A_*(X_1)| - |A_*(X_1) \cap A^*(X_2)|}{|A^*(X_1)| - |A^*(X_1) \cap A_*(X_2)|} \end{aligned}$$

- $|A^*(X_2)| \leq |A_*(X_1)|$. Under this situation, we have $|A_*(X_2)| \leq |A^*(X_1)|$. Therefore

  $$R^\circ_A(X) \leq 1 - \frac{|A_*(X_1)| - |A^*(X_2)|}{|A^*(X_1)|}$$

  Considering $R^\circ_A(X_1) = 1 - \frac{|A_*(X_1)|}{|A^*(X_1)|}$ and assume $s^* = \frac{|A^*(X_2)|}{|A^*(X_1)|}$, we have

  $$R^\circ_A(X) \leq R^\circ_A(X_1) + s^*$$

  Similarly, let $s_* = \frac{|A_*(X_2)|}{|A_*(X_1)|}$, we can get the lower limit:

  $$\begin{aligned} R^\circ_A(X) &\geq 1 - \frac{|A_*(X_1)|}{|A^*(X_1)| - |A_*(X_2)|} \\ &\geq \frac{R^\circ_A(X_1) + (R^\circ_A(X_1) - 1)s_*}{1 + (R^\circ_A(X_1) - 1)s_*} \end{aligned}$$

- $|A^*(X_2)| \geq |A_*(X_1)|$ and $|A_*(X_2)| \leq |A^*(X_1)|$. For this case, we have $\max\{|A^*(X_2) \cap A_*(X_1)|\} = |A_*(X_1)|$ and $\max\{|A_*(X_2) \cap A^*(X_1)|\} = |A_*(X_2|)$. Thus

  $$\begin{aligned} R^\circ_A(X) &\leq 1 - \frac{|A_*(X_1)| - |A_*(X_1)|}{|A^*(X_1)|} = 1 \\ R^\circ_A(X) &\geq 1 - \frac{|A_*(X_1)|}{|A^*(X_1)| - |A_*(X_2)|} \\ &\geq \frac{R^\circ_A(X_1) + (R^\circ_A(X_1) - 1)s_*}{1 + (R^\circ_A(X_1) - 1)s_*} \end{aligned}$$

- $|A^*(X_2)| \geq |A_*(X_1)|$ and $|A_*(X_2)| \geq |A^*(X_1)|$. Similar to last case, we have $\max\{|A^*(X_2) \cap A_*(X_1)|\} = |A_*(X_1)|$ and $\max\{|A_*(X_2) \cap A^*(X_1)|\} = |A^*(X_1)|$. Then, we have $R^\circ_A(X) \leq 1 - \frac{|A_*(X_1)| - |A_*(X_1)|}{|A^*(X_1)|} = 1$. For the lower limit, we have $R^\circ_A(X) \geq 1 - \frac{|A_*(X_1)|}{|A^*(X_1)| - |A^*(X_1)|}$. Obviously, the roughness of approximation does not exist under this situation. This is because $|A_*(X_1)|$ is less than $|A^*(X_2)|$, and the whole set X_1 has been removed in the extreme situation. If $|A_*(X_2) \cap A^*(X_1)| = |A_*(X_1)|$, then we have $R^\circ_A(X) \geq 0$.

Obviously, a corresponding conclusion exists for accuracy of approximation.

***Corollary* 4** *The accuracy of approximation of the difference set of two rough sets satisfies the following rules*

- *If* $|A^*(X_2)| \leq |A_*(X_1)|$, *then*

$$\frac{\alpha_A(X_1)}{1 - \alpha_A(X_1)s_*} \geq \alpha_A(X) \geq \alpha_A(X_1) - s^*$$

- *If* $|A^*(X_2) \geq |A_*(X_1)|$ *and* $|A_*(X_2)| \geq |A^*(X_1)|$, *then*

$$\frac{\alpha_A(X_1)}{1 - \alpha_A(X_1)s_*} \geq \alpha_A(X) \geq 0$$

- *if* $|A^*(X_2)| \geq |A_*(X_1)|$ *and* $|A_*(X_2)| \geq |A^*(X_1)|$, *then the accuracy of approximation of operands can not bound the accuracy of approximation for the difference set under this situation*

where, $s_* = \frac{|A_*(X_2)|}{|A_*(X_1)|}$ *and* $s^* = \frac{|A^*(X_2)|}{|A^*(X_1)|}$.

4.6.4 Complement of rough set

The upper and lower limits of complement set can be derived from the difference set.

***Theorem* 3** *The roughness of approximation for a complement set of a rough set can be represented as*

$$0 \leq R^\circ_A(\overline{X}) \leq s$$

where s is the ratio between the cardinalities of the B-upper approximations of X and U: $s = \frac{|A^*(X)|}{|A^*(U)|}$.

Proof U is a universe, we have $X \in U$ and $A_*(U) = A^*(U)$ and $R^\circ_A(U) = 0$ then $|A^*(X)| \leq |A_*(U)|$ Let $X_1 = X$, $X_2 = U$, $s = \frac{|A^*(X)|}{|A^*(U)|}$ and $t = \frac{|A_*(X)|}{|A_*(U)|}$

according to the theorem for a difference set, we have $R^{\circ}_{A}(\overline{X}) \leq R^{\circ}_{A}(U) + s = s$.

$$R^{\circ}_{A}(\overline{X}) \geq \frac{R^{\circ}_{A}(U) + (R^{\circ}_{A}(U) - 1)t}{1 + (R^{\circ}_{A}(U) - 1)t} = -\frac{t}{1-t}$$

Because $X \in U$, so $0 \leq t \leq 1$, therefore we have $R^{\circ}_{A}(\overline{X}) \geq 0$. We know that for any $R^{\circ}_{A}(\overline{X})$, $R^{\circ}_{A}(\overline{X}) \geq 0$, then we have $R^{\circ}_{A}(\overline{X}) \geq \max\{0, -\frac{t}{1-t}\} \geq 0$.

For accuracy of approximation, we have similar conclusion.

***Corollary* 5** *The accuracy of approximation for a complement set of a rough set can be represented as*

$$1 \geq \alpha_{A}(\overline{X}) \geq 1 - s$$

where s is the ratio between the cardinalities of the B-upper approximations of X and U: $s = \frac{|A^{*}(X)|}{|A^{*}(U)|}$.

Roughness is an important indicator for the uncertainty associated with a rough set. It propagates through various set operations and influences the accuracy of the results of set operations. We proved that there is no defined bound for the intersection operation of any two rough sets but there are bounds for union, complement and most difference operations. The results show that we can get some indication of the roughness or accuracy of the resulting rough sets before completing an operation involving two or more large rough sets. This is beneficial for decision making involving large volume of rough set operations.

4.7 Grey sets and grey geometry

4.7.1 Grey sets

There are two particular uncertainties associated with sets:

- We may have a concept that is imprecise or uncertain. In this case the concept itself is fuzzy.
- The elements of the sets are uncertain.

The fuzzy concepts are caused by our ambiguous description, which is a perception uncertainty. However, the uncertain element is a result of the uncertain status or the incompleteness of information on the status of the element. In this situation, the concept of the set is clear, but the status of an element is undetermined. A third type of uncertainty is a combination of both uncertainties, which is more common in the real world.

Many models have been developed in an attempt to tackle uncertainties associated with sets, such as probability [21], fuzzy sets [232] and rough sets

[138]. The concept of probability is closely related to set operations, and reflects the randomness of an event. However, it does not reveal the aforementioned fuzzy concepts and uncertain position of elements. Fuzzy sets and rough sets are the two mainstream models in set uncertainty. Some methodologies merging these two models have been proposed, such as the fuzzy rough and rough fuzzy sets [51, 130, 231]. This section presents a new approach to unifying fuzzy sets and rough sets through the notion of a grey set, based on grey systems. The advantage of this approach is that in this one model we can discuss the various uncertainties more clearly.

Fuzzy sets and rough sets cover different uncertainties in sets, thus being complementary to each other. Fuzzy sets concern the fuzziness of concepts caused by human perception, it does not focus on the unknown status of individual elements caused by incomplete information. Some extended fuzzy sets have involved incomplete information in some sense, like interval-valued fuzzy sets. As extensions of fuzzy sets, they have close relationships with fuzziness. On the contrary, rough sets are much more powerful in dealing with incompleteness of information rather than fuzziness. Rough sets clearly separate those elements with clear status from the elements with lack of information by means of its set approximation and equivalence relation. Therefore, a combination of the two models is an attractive direction for research. There have been many research on this subjects, such as the fuzzy rough sets [51, 130, 231].

However, the uncertainty caused by incomplete information has not yet been fully investigated although rough sets have made some progress. Rough sets mainly deal with elements which have only three value logic: YES, NO or UNKNOWN. There is no consideration for something in between like the fuzzy concepts. The introduction of tolerance instead of equivalence improves it [106], but a more general model is necessary to describe the information incompleteness both for rough sets and fuzzy sets. In addition to this, the uncertainty of a set as a whole has not been investigated as much as individual elements. A general model for set uncertainty is a necessary component in uncertainty models for sets. Here, we introduce the degree of greyness from grey system into set uncertainty, and define a generalised set – grey set. The next Section firstly provides an overview of grey numbers and grey systems.

Both traditional crisp sets and fuzzy sets need a clear defined membership or characteristic function value. Rough sets have a similar real number membership function like fuzzy sets[106]. However, this clear defined number is difficult to know in some situation. If a human brain has a "fuzzy" concept, how can we ask the human brain to give a clear defined value for a fuzzy perception? Interval valued fuzzy sets have successfully expressed this situation in the case of fuzzy sets, and we will extend this to a more general level using grey numbers. Similar to grey numbers, we classify sets into three categories: white sets, grey sets and black sets. For the sake of simplicity, we limit the value of a set character function within [0,1].

Before the definition of grey sets, we extend the definition of grey numbers to the discrete situation. The grey numbers discussed so far are concerned only with continuous numbers. For instance, the underlying white number for a grey number [0.2,0.8] can be any real number v satisfying $0.2 \leq v \leq 0.8$. There are potentially infinite options satisfying this condition. However, there are situations that the white number can only have limited options within its boundary. As an example, we know a white number w could be any one among 0.2, 0.5 and 0.8, but it can not be any other number. That is to say, for instance, 0.3 is not an option at all. Traditional grey systems or interval values do not include this situation. Here, we extend the definition of grey numbers to discrete situations.

***Definition* 17 (Discrete grey numbers)** *A discrete grey number is a number with clear upper and lower boundaries but which has limited options inside its boundary.*

A discrete grey number $v^{\pm}$ can be expressed as follow:

$$v^{\pm} = \{v^{-}, v_1, v_2, \ldots, v_k, v^{+}\}$$

where, $v^{-} \leq v_1 \leq v_2 \leq \ldots \leq v_k \leq v^{+}$ and $0 \leq k < \infty$.

Corresponding to discrete grey numbers, the grey numbers introduced in last section are called continuous grey numbers. Similar to continuous grey numbers, a discrete grey number represents an underlying white number. The underlying white number expressed by a discrete grey number can only be one and only one of the limited candidates inside its boundary. However, the underlying white number expressed by a continuous grey number has infinite options because of its continuous domain. Without specific notation, we call both grey numbers in this paper. A set of grey numbers was called grey set in [214], and we call it grey number set here to differentiate it from the grey sets defined in this paper. The grey number set with lower and upper limits n^{-} and n^{+} is represented as $[n^{-}, n^{+}]^{\pm}$ in this paper. A grey number set within 0 and 1 would be $[0,1]^{\pm}$, which is clearly different from a real number set $[0, 1]$.

***Definition* 18 (White set)** *For a set $W \subseteq U$, if the characteristic function value of each e with respect to W can be expressed with a single white number $v \in V$*

$$\chi_W : U \rightarrow V$$

then W is a white set.

In fact, the white sets are the same as type-1 fuzzy sets, here we call it white sets to differentiate it from other extended fuzzy sets.

***Definition* 19 (Black set)** *For a set $B \subseteq U$, if the characteristic function value of each e with respect to B can only be expressed with a grey number $b^{\pm} = [0,1]$ or $b^{\pm} = \{0, v_1, v_2, \ldots, 1\}$*

$$\chi_B : U \to \{b^{\pm}\}$$

then B is a black set. Here, $0 < v_1 \leq v_2 \leq \ldots < 1$

***Definition* 20 (Grey set)** *$V^{\pm} = [0,1]^{\pm}$ is a grey number set. For a set $G \subseteq U$, if the characteristic function value of e with respect to G can be expressed with a grey number $v^{\pm} = [v^-, v^+] \in V^{\pm}$ or $v^{\pm} = \{v^-, v_1, v_2, \ldots, v_k, v^+\} \in V^{\pm}$*

$$\chi_G : U \to V^{\pm}$$

then G is a grey set.

Similar to the expression of a fuzzy set, a grey set G is represented with its relevant elements and their associated grey number for characteristic function:

$$G = {v_1}^{\pm}/e_1 + {v_2}^{\pm}/e_2 + \ldots + {v_n}^{\pm}/e_n$$

The characteristic function here is a general expression, it does not exclude any relevant criteria in defining a set. Therefore, it can be replaced by a probability function, membership function, possibility function etc. For a white set, we know clearly the relationship between an element and a set. Obviously, a white set here is different from a crisp set in traditional sets. A white set has a clear relationship between the set and relevant elements, and that relationship is not necessarily a crisp relationship. If we replace the characteristic function with fuzzy membership function, then the white set become a standard type-1 fuzzy set.

Student evaluation There are five students in table 4.1, their name, gender, working attitude and exam results are listed in the table. A set A for evaluating the study of students is to be established with respect to different attributes. Assume a_i is a student in the table, and $i = 1, 2, 3, \ldots, n$. n is the number of students. We can get A directly from Exam Result attribute in the table, and it is also possible to establish A indirectly using other attributes, such as Working Attitude and Gender. Here, we adopt Working Attitude to establish a grey set.

Name	Gender	Working Attitude	Exam Result
Mike	Male	Good	Good
Jane	Female	Neutral	Good
Claire	Female	Neutral	Neutral
David	Male	Neutral	Poor
Lisa	Female	Poor	Poor

Table 4.1: Information for 5 people

The exam result shows some kind of relationships with Working Attitude. A characteristic function is established according to the relationship between Working Attitude and Exam Result:

$$f_A{}^c(a_i) = \begin{cases} 1 & \text{if } a_i\text{'s Working attitude = good;} \\ [0,1] & \text{if } a_i\text{'s Working attitude = neutral;} \\ 0 & \text{if } a_i\text{'s Working attitude = poor.} \end{cases}$$

Under this characteristic function, $A = [1,1]/Mike + [0,1]/Jane + [0,1]/Claire + [0,1]/David + [0,0]/Lisa = 1/Mike + [0,1]/Jane + [0,1]/Claire + [0,1]/David + 0/Lisa$. Obviously, A is a grey set.

It is clear that a grey set has ill defined relationships between some elements and the set, and their characteristic functions have a grey number for a given attribute value.

From this example, it is clear that there are two different kinds of students in a grey set A according to their characteristic function values: students with white numbers (0 or 1) and students with grey numbers. They are two different categories. We can classify the elements relevant to a grey set into three different categories: white, grey and black elements.

***Definition* 21 (White element)** *U is the universe of discourse, G is a grey set and $G \subseteq U$. e is an element relevant to G and $e \in U$. $v^{\pm}$ is the value for characteristic function of e with respect to G. If $v^- = v^+$, then e is called a white element*

***Definition* 22 (Grey element)** *U is the universe of discourse, G is a grey set and $G \subseteq U$. e is an element relevant to G and $e \in U$. $v^{\pm}$ is the value for characteristic function of e with respect to G. If $v^- \neq v^+$, then e is called a grey element*

***Definition* 23 (Black element)** *U is the universe of discourse, G is a grey set and $G \subseteq U$. e is an element relevant to G and $e \in U$. $v^{\pm}$ is the value for characteristic function of e with respect to G. If $v^- = 0$ and $v^+ = 1$, then e is called a black element*

Because of the existence of grey and black elements, the relationships between some elements and a grey set are not completely known. The value for its corresponding characteristic function can only be expressed as a grey number. This is caused by the incomplete information of this element. Similar to the case for a grey number, the uncertainty caused by the information incompleteness can be measured using a degree of greyness. Considering the specific feature of grey sets, the degree of greyness for an element and a set are defined here.

***Definition* 24 (Degree of greyness for an element)** *U is the finite universe of discourse, e is an element and $e \in U$. For a grey set $G \subseteq U$, the characteristic function value of e with respect to G is $v^{\pm} = [v^{-}, v^{+}]$ or $v^{\pm} = \{v^{-}, v_1, v_2, \ldots, v_k, v^{+}\}$. The degree of greyness $g^{\circ}{}_G(e)$ of element e for set G is expressed as*

$$g^{\circ}{}_G(e) = \frac{v^{+} - v^{-}}{v^{max} - v^{min}}$$

Here,v^{max} and v^{min} are the maximum value and minimum value of characteristic function, $0 \leq v^{-} \leq v_1 \leq v_2 \leq \ldots \leq v_k \leq v^{+} \leq 1$ and $0 \leq k < \infty$.

According to our definition for grey sets, we have

$$v^{max} = 1$$

and

$$v^{min} = 0$$

therefore

$$g^{\circ}{}_G(e) = v^{+} - v^{-}$$

Based on the degree of greyness for an element, a degree of greyness for a set is defined as follow.

***Definition* 25 (Degree of greyness for a set)** *U is the finite universe of discourse, G is a grey set and $G \subseteq U$. e_i is a an element relevant to G and $e_i \in U$. $i = 1, 2, 3, \ldots, n$ and n is the cardinality of G. The degree of greyness of set G is defined as*

$$g^{\circ}{}_G = \frac{\sum g^{\circ}{}_G(e_i)}{n}$$

Example for degree of greyness According to the given definition, the uncertainty caused by incomplete information for the evaluation of students under different considering attributes can be measured using the degree of greyness for the elements and sets. Considering data in table 4.1, we evaluate their results with respect to the three different attributes and setup three sets: $A = \{$evaluation using working attitude$\}$, $B = \{$evaluation using exam results$\}$

and $C = \{\text{evaluation using gender}\}$. It is obvious that B set is a white set where the result is evaluated by itself. For C set, one can not know if a male (female) has a good or poor result because that both results exist in table 4.1 for male (female). Therefore, C is a black set. From our previous analysis, A is a grey set. Therefore, we have

$$A = 1/Mike + [0,1]/Jane + [0,1]/Claire + [0,1]/David + 0/Lisa$$

$$B = 0/Mike + 0/Jane + 0/Claire + 0/David + 0/Lisa$$

$$C = 1/Mike + 1/Jane + 1/Claire + 1/David + 1/Lisa$$

The degree of greyness of each element in the relevant set is derived from their characteristic functions. For instance, the degree of greyness for Jane could be calculated as

$$g^{\circ}_{A}(Jane) = 1 - 0 = 1$$

$$g^{\circ}_{B}(Jane) = 0 - 0 = 0$$

$$g^{\circ}_{C}(Jane) = 1 - 1 = 0$$

For the grey set A derived from Working attitude, its degree of greyness is

$$g^{\circ}{}_{A} = \frac{0+1+1+1+0}{5} = 0.6$$

The results for the sets evaluated according to Exam Result, Working attitude and Gender are shown in Table 4.2.

Name	Exam Result		Working attitude		Gender	
	Element	Set	Element	Set	Element	Set
Mike	0	0	0	0.6	1	1
Jane	0		1		1	
Claire	0		1		1	
David	0		1		1	
Lisa	0		0		1	

Table 4.2: Example for degree of greyness

From Table 4.2, it is clear that a white set has a degree of greyness of 0, a black set has a degree of greyness of 1 and a grey set has a degree of greyness between 0 and 1.

Theorem 4 *U is the finite universe of discourse, G is a grey set and $G \subseteq U$. e is an element and $e \in U$. $v^{\pm}{}_{e}$ is a value for the characteristic function with respect to e. $g^{\circ}{}_{G}(e)$ is the degree of greyness of e, and $g^{\circ}{}_{G}$ is the degree of* *greyness for G. The following properties hold for e and G:*

- *G is a white set iff* $g^{\circ}{}_G = 0$
- *G is a black set iff* $g^{\circ}{}_G = 1$
- *G is a crisp set iff* $g^{\circ}{}_G = 0$ *and* $v^{\pm}{}_e \in \{0,1\}$ *for any* $e \in U$
- *G is a type-1 fuzzy set iff* $g^{\circ}{}_G = 0$ *and* $v^{\pm}{}_e \in [0,1]$ *for any* $e \in U$
- *G is an interval-valued fuzzy set iff* $v^{\pm}{}_e$ *is a continuous grey number for any* $e \in U$

Proof Assume the value of characteristic function as $v_i{}^{\pm}$. i is the index of the element. If $g^{\circ}{}_G = 0$, then $\sum g^{\circ}{}_G(e_i) = 0$. We have $g^{\circ}{}_G(e_i) = 0$ for any i, and therefore $v_i{}^- = v_i{}^+$ for any i. Thus v_i is a white number for any i, and G is a white set. If G is a white set, then v_i is a white number for any i and $v_i{}^- = v_i{}^+$. Thus $g^{\circ}{}_G(e_i) = 0$ for any i, and $\sum g^{\circ}{}_G(e_i) = 0$. Therefore $g^{\circ}{}_G = 0$. The second conclusion can be proved in a similar way.

From the first rule in this theorem, G is a white set when $g^{\circ}{}_G = 0$. Then $v_i{}^{\pm} = v \in [0,1]$. Obviously, v is a white number between 0 and 1. Therefore, G is a type-1 fuzzy set. If we know G is a type-1 fuzzy set, then its fuzzy membership has a clear defined white number between 0 and 1 as its value, then $g^{\circ}{}_G(e) \in [0,1]$ for any $e \in U$. Therefore its degree of greyness for each element is 0, then we know $g^{\circ}{}_G = 0$.

The third rule is straight away. If we have $g^{\circ}{}_G = 0$ then G is a white set. For a white set, if $v^{\pm}{}_e \in \{0,1\}$, then its characteristic function would only have a value of 0 or 1. It is clear that G is a crisp set. If G is a crisp set, it is clear that $v^{\pm}{}_e \in \{0,1\}$ and $g^{\circ}{}_G = 0$ hold. Similarly we can prove the fourth conclusion.

If we consider characteristic function as fuzzy membership function μ and $V^{\pm} = [0,1]^{\pm}$, then we have

$$\mu_G : U \to V^{\pm}$$

Under continuous grey number, $v^{\pm} = [v^-, v^+] \subseteq V^{\pm}$ can be considered as an interval d and $d \in D[0,1]$. Here $D[0,1]$ represents an interval between 0 and 1. Hence the grey set G can be expressed as

$$\mu_G : U \to D[0,1]$$

this is an interval valued fuzz set. For an interval valued fuzzy set, the fuzzy membership μ can be considered as characteristic function, then we have

$$\chi{:}U \to D[0,1]$$

Assume the membership interval $d = [v^-, v^+]$, which is a continuous interval. Represent this continuous interval with a continuous grey number $v^\pm = d = [v^-, v^+]$, then we have a grey set with μ as characteristic function.

$$\mu{:}U \to V^\pm$$

This theorem shows that grey sets extend crisp sets, fuzzy sets and interval-valued fuzzy sets.

Theorem **5** *G is a rough set iff $g^\circ{}_G > 0$ and $v^\pm{}_e \subseteq \{0,1\}$ holds for any $e \in U$. Here, U is the finite universe of discourse, G is a grey set and $G \subseteq U$. e is an element and $e \in U$. $v^\pm{}_e$ is a value for the characteristic function with respect to e. $g^\circ{}_G$ is the degree of greyness for G.*

Proof If $v^\pm{}_e \subseteq \{0,1\}$ holds for any $e \in U$, then $v^\pm{}_e$ is a discrete grey number, and $v^\pm{}_e = \{0,1\}$. There are only three options for the result value of $v^\pm{}_e$: 0, 1 or $\{0,1\}$. The elements in U can be classified into three different crisp sets according to the value of $v^\pm{}_e$: G_* for $v^\pm{}_e = 1$, F for $v^\pm{}_e = \{0,1\}$ and $\sim G$ for $v^\pm{}_e = 0$. Obviously, $G_* \cap \sim G = \Phi$. If $g^\circ{}_G > 0$, then $F \neq \Phi$. The elements in F are not determined, and they may belong to G_* or $\sim G$ with more information. There are two possible extreme situations: $v^\pm{}_e = 1$ for each $e \in F$ or $v^\pm{}_e = 0$ for each $e \in F$. For the first situation, we get the maximum $G^* = G_* \cup F$. For the second situation, we get the minimum G_*. Obviously, $G^* \supseteq G \supseteq G_*$. Let $R = U \times U$ be an equivalence relation on the universe U, $[e]_R$ is the equivalence class containing e. Thus each $e \in U$ represents an equivalent class $[e]_R$. The elements in $[e]_R$ should include all validate characteristic function values for $e \in U$. From our analysis of G_* and G^*, we have

- $v^\pm{}_{e_i} = 1$ if $e_i \in [e]_R$ and $[e]_R \subseteq G_*$
- There is at least one element e_i satisfying $v^\pm{}_{e_i} = 1$ for each $e_i \in [e]_R$ if $[e]_R \subseteq G^*$

Here, i is the index of the elements in $[e]_R$, $i = 1, 2, \ldots, k$. k is the number of the elements in $[e]_R$. Therefore, we have

- $[e]_R \subseteq G$ iff $e \in G_*$
- $[e]_R \cap G \neq \Phi$ if $e \in G^*$

Then we have the lower approximation $R_*(G)$ and the upper approximation $R^*(G)$ as follows:

- $R_*(G) = \{e \in U | [e]_R \subseteq G\} = G_*$
- $R^*(G) = \{e \in U | [e]_R \cap G \neq \Phi\} = G^*$

Obviously, under the given condition, a grey set is equivalent to a rough set. For a rough set G, it satisfies the two equations above. A characteristic function could be established:

$$f_G{}^c(e) = \begin{cases} 1 & \text{if } e \in R_*(G); \\ \{0,1\} & \text{if } e \in R^*(G) \text{ but } e \notin R_*(G); \\ 0 & \text{if } e \notin R^*(G). \end{cases}$$

Obviously, the value of this characteristic function contains discrete grey number. Assume $V^{\pm} = \{0,1\}$, then G satisfies

$$\chi_G : U \rightarrow V^{\pm}$$

This is a grey set.

This theorem proves that grey sets include rough sets as a special case.

There are many models for uncertainty representation, but the main stream methodologies at present are fuzzy sets and rough sets. They represent different aspects of uncertainties, and provide complementary functions in uncertainty modelling. There have been considerable efforts in unifying them by means of fuzzy rough or rough fuzzy models. However, we present a different route to this unification: a grey model. We propose grey sets unifying fuzzy sets and rough sets in a simple model. Our results show that a grey model can be specified to both fuzzy sets and rough sets.

4.7.2 Grey geometry

As a mathematical foundation of numerous geometrical operations, computational geometry is playing a significant role in the development of areas like computer graphics [7, 67], computer aided design [191, 192], Geographical Information Systems (GIS) [109, 72, 70] etc. However, the complexity of the real world raises also many challenges to the further application of computational geometry. One of these is the representation of uncertainties in geometrical objects [73, 200]. Because of various limits in measurement, data in the real world are sometimes not so accurate in the sense of mathematics. There may be different errors involved in the measuring operations, such as the errors from equipment, the errors from human operations etc. In the same time, the accuracy is a relative concept, and it is subject to changes under different scales. For instance, the distance measurement may be accurate in the scale of Kilometres, but it may contain a significant unknown for Millimetre scale. It is very common in engineering that the measured data contains some errors, and thus the geometrical object can not be kept strictly in the sense of traditional geometry. This problem is not so obvious in traditional application areas like computer graphics or computer aided design for small size entities. However, its existence significantly influences the application of computational geometry in new application areas where a huge amount of large scale geometrical objects are concerned, such as GIS [109]. There are many operations sensitive to errors and other uncertainties in GIS, such as data structure tuning, generalisation, topology, overlay, spatial accuracy and analysis [36]. The traditional geometry does not consider these uncertainties at all, and hence its application in problems involving large amount of uncertainties is limited. Some methodologies have appeared in the application side, such as the buffer zone approach proposed in GIS research [200]. To embrace the challenge of computing applications, a systematic investigation into the representation of uncertainties, especially errors in measuring data, is essential. In the late 70's, Azriel Rosenfeld introduced fuzzy geometry to consider fuzziness in geometry [150], which is then applied into to image processing [135]. Fuzzy logic is the most popular technique in dealing with uncertainties related with human perception. Whereas, the wide existence of measuring errors in spatial data involves not only human perceptions, but also the limitation of equipment and other physical limitations. This kind of uncertainty is different from the human perceptions in that they have clear boundaries but unknown positions within its boundaries. It demonstrates our limited or imperfect knowledge of the geometrical features of the real world objects, hence a methodology specific for this kind of uncertainty is needed for a further contribution of computational geometry to the computing industry. Here, we define a new type of uncertainty representation for geometry using grey systems [47], an emerging theory to treat with imperfect information, as a step forward to enable geometry to support spatial analysis with uncertainties.

By grey geometry, we mean the geometry where every object is considered as a grey object. Different from objects in traditional geometry, objects in grey geometry have no clearly defined positions or boundaries but a known scope of possible values. This is different from fuzzy geometry in that a grey object has clear definition of its scope, and it can move around only within that scope. It is clearly defined in the sense of its scope, but not determined with respect to its exact position or boundaries. As in the case of traditional geometry, the first thing for grey geometry is to define its elements which constitute the grey geometrical objects.

Grey points

The basic element for a geometrical operation is a point, and it is the same for grey geometry. Different from traditional geometry, a 2D point in grey geometry is not strictly a "point" in the sense of traditional geometry. It is some kind of "area" where its position is bounded. The comparison between traditional point and grey point is shown in Figure 4.13. Figure 4.13(a) is a point in traditional geometry, and Figure 4.13(b) represents a grey point in grey geometry.

***Definition* 26 (Grey point)** *For a given grey number* $x^{\pm} = [x_1, x_2]$ *in X dimension, and* $y^{\pm} = [y_1, y_2]$ *in Y dimension, a grey point* $P(x^{\pm}, y^{\pm})$ *is represented as*

$$P(x^{\pm}, y^{\pm}) = P([x_1, x_2], [y_1, y_2])$$

A grey point $P(x^{\pm}, y^{\pm})$ represents a point $p(x, y)$ where $x \in [x_1, x_2]$ and $y \in [y_1, y_2]$.

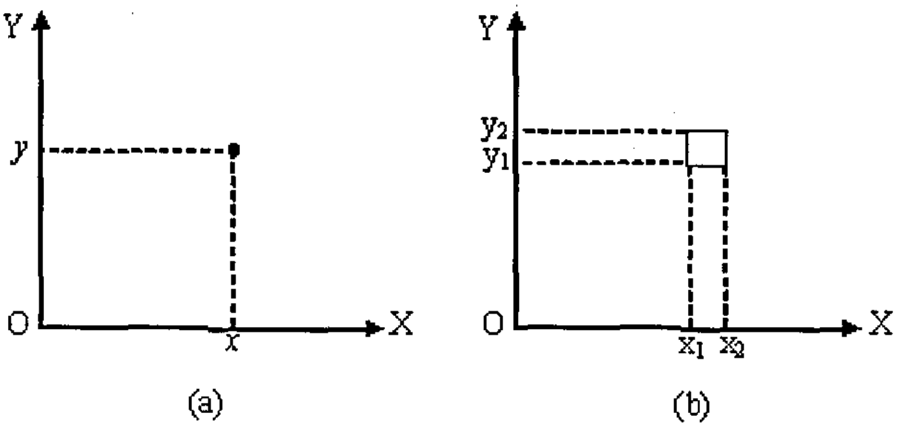


Figure 4.13: Comparison between traditional points and grey points

Figure 4.13(a) is a point in traditional geometry, and Figure 4.13(b) represents a grey point in grey geometry. Obviously, a grey point is not an exact position of point, and it is in fact a bounded area or volume where the point exists.

Grey lines and grey segments

Similar to traditional geometry, we define the grey line as the "line" passing through two grey points. It consists of all the collinear grey points on this "line". More accurately, a grey line is defined in the form of its grey parameters in analytic geometry.

***Definition* 27 (Grey line)** *: For given grey numbers $a^{\pm} = [a_1, a_2]$, $b^{\pm} = [b_1, b_2]$ and $c^{\pm} = [c_1, c_2]$, a grey line is defined as*

$$a^{\pm}x + b^{\pm}y + c^{\pm} = 0$$

where, x and y are the coordinates of the two dimensions. Obviously, the value of x and y would also be grey values.

Similar to line segment in traditional geometry, a grey segment is a closed subset contained between two grey end points on a grey line. Figure 4.14 demonstrates a 2D grey segment between grey point P1 and P2. From Figure 4.14, it is clear that grey segment defines an area where the segment may fall in, but the exact position of the segment is unknown. Both the length and direction are grey values and they have upper and lower limits as other grey numbers.

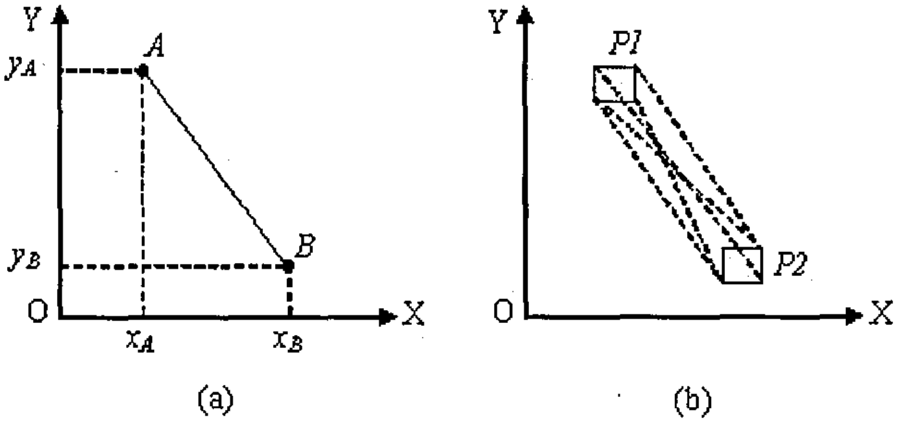


Figure 4.14: Comparison between traditional line segments and grey line segments

Grey Polygons

In computational geometry, the most important shape is a polygon. It is the basis for modern computer visualisation and graphics. Its feature is also the most important content in grey geometry.

In traditional geometry, "a polygon is the region of a plane bounded by a finite collection of line segments forming a simple closed curve" [134]. Similarly, a grey polygon is a grey region of a plane bounded by a finite collection of grey line segments forming a simple closed grey curve.

***Definition* 28 (Grey polygons)** *: Let $P_1, P_2, , P_n$ be n grey points in a plane, and $S_1 = P_1P_2, S_2 = P_2P_3, \ldots, S_{n-1} = P_{n-1}P_n, S_n = P_nP_1$ be n grey segments connecting the grey points, then these segments bound a polygon iff*

- *The intersection of each pair of grey segments adjacent in the cyclic ordering is a single grey point shared between them:* $S_i \cap S_{i+1} = P_{i+1}$*, for all* $i = 1, 2, \ldots, n$ *(we define* $n + 1 = 1$*)*
- *Nonadjacent segments do not intersect:* $S_i \cap S_j = \emptyset$ *, for all* $j \neq i + 1$.

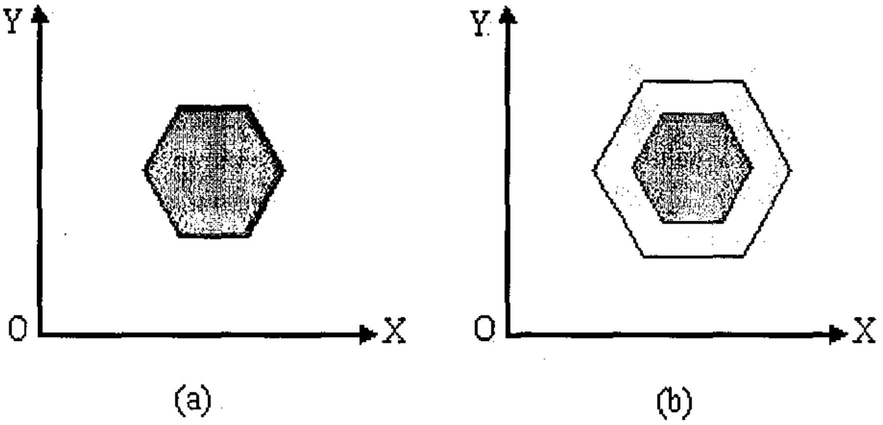


Figure 4.15: Comparison between traditional polygons and grey polygons

The comparison between grey polygon and traditional polygon is illustrated in Figure 3. It shows a grey polygon with regular inner and outer boundaries. However, the inner and outer boundaries may not even be similar shapes. The inner boundary defines an area where the points definitely belong to the polygon, but the area between inner and outer boundaries refers to points which may or may not belong to the polygon. It is the so called grey area.

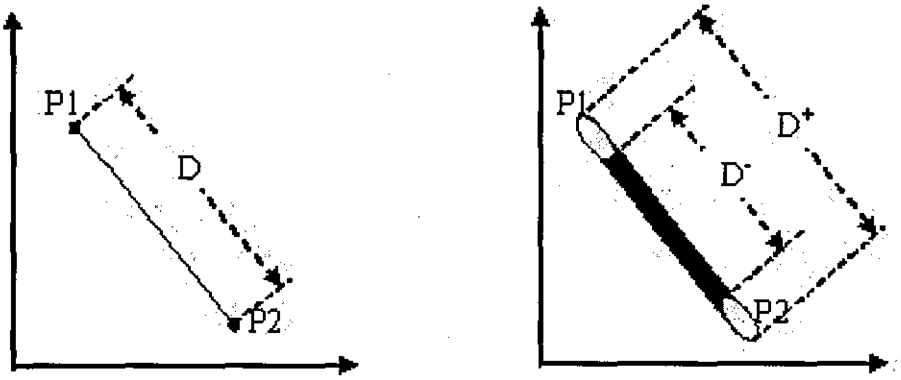
Grey distances

In traditional geometry, distance is the shortest length of path from one point to another point. It is in fact the length of line segment passing through the two points, as shown in Figure 4.16(a). In grey geometry, the distance between two grey points is the length of grey line segment passing through the two grey points, as illustrated in Figure 4.16(b).

***Definition* 29 (Grey distance)** *: For two given grey points* $P_1([x_{11}, x_{12}], [y_{11}, y_{12}])$ *and* $P_2([x_{21}, x_{22}], [y_{21}, y_{22}])$*, their grey distance is a grey value* D

$$D = [D^-, D^+]$$

where $D^- = \sqrt{(x_{21} - x_{12})^2 + (y_{21} - y_{12})^2}$ *and* $D^+ = \sqrt{(x_{22} - x_{11})^2 + (y_{22} - y_{11})^2}$



(a) (b)

Figure 4.16: Comparison between traditional distances and grey distances

Degree of greyness for 2D grey objects

In geometry, both lines and polygons consist of points. A line segment can be considered as a set of points distributing on the segment. Similarly, a polygon is a set of points bounded within the boundary of the polygon. A line segment or a polygon could be considered as sets consisting of points. Therefore, a grey point, a grey line segment or a grey polygon could be considered as grey sets of points. In these sets, each point is associated with a degree of greyness: $0 \leq g_i^\circ \leq 0$.

In most cases, it is very difficult to quantify the membership of each individual points of a geometrical object. However, it is it is possible for us to know which part has a certain relationship with the object and which part has only unknown information. In this case, we take a discrete grey number $\{0, 1\}$ as their unknown characteristic function value of elements. For those known elements, their characteristic function value would be 1 or 0 depending on if they are in the object or not. From definition 24, we have the properties of degree of greyness of geometrical objects:

- If P is a grey point, p is a point and $p \in P$, then $g^\circ(p) = 1$;
- If P_1P_2 is a grey line segment (P_1 and P_2 are the two end grey points), p is a point and $p \in P_1P_2$, then we have

$$g^\circ(p) = \begin{cases} 1 & \text{if } p \in P_1 \cup P_2 \\ 0 & \text{if } p \in P_1P_2 \text{ and } p \notin P_1 \cup P_2 \end{cases}$$

- If P is a grey polygon, p is a point and $p \in P$. $P_1, P_2, \ldots, P_n$ are its grey vertexes and $P_1P_2, P2p3, \ldots, P_nP_1$ are its grey boundaries (segments).

The degree of greyness of element p is

$$g^{\circ}(p) = \begin{cases} 1 & \text{if } p \notin P_1P_2 \cup P_2P_3 \cup \ldots \cup P_{n-1}P_n \cup P_nP_1 \\ 0 & \text{if } p \in P_1P_2 \cup P_2P_3 \cup \ldots \cup P_{n-1}P_n \cup P_nP_1 \end{cases}$$

With the value of degree of greyness for each element, we now can calculate the degree of greyness of each set according to Definition 25:

- Grey points P: $g^{\circ}_G(P) = \frac{\sum_{i=1}^{m_p} 1}{n_p}$;
- Grey segments P_1P_2: $g^{\circ}_G(P_1P_2) = \frac{\sum_{i=1}^{m_l} 1}{n_l}$;
- Grey polygons P: $g^{\circ}_G(P) = \frac{\sum_{i=1}^{m_P} 1}{n_P}$

Where n_p, n_l and n_P are the total number of points in a grey point, a grey segment or a grey polygon, m_p, m_l and m_P are the number of points in the unknown (boundary) region of objects. Obviously, these parameters vary with the change of the resolution of the object representation. If the resolution is fine enough, then the ratio of $\frac{m_p}{n_p}$, $\frac{m_l}{n_l}$ and $\frac{m_P}{n_P}$ would converge to 1, $\frac{l_B}{l_L}$ and $\frac{S_B}{S_P}$. Here, l_B and l_L refer to the boundary length of a grey segment (the total length of the two grey end points and the total length of the whole grey segment. Similarly, S_B denotes the area of boundary region (grey segments), and S_P represents the whole area of the grey polygon P.

Therefore, we have

- Grey points P: $g^{\circ}_G(P) = 1$;
- Grey segments P_1P_2: $g^{\circ}_G(P_1P_2) = \frac{l_B}{l_L}$;
- Grey polygons P: $g^{\circ}_G(P) = \frac{S_B}{S_P}$

Degree of greyness of a set indicates the uncertainty associated with the set. Therefore, it is a good indicator for uncertainty control. Under the given assumption here for geometry, we consider the characteristic function value as three categories: 0, 1 or $\{0, 1\}$. Obviously, from Theorem 5 we know that a grey set is equivalent to a rough set. From Definition 24 and Definition 6, the degree of greyness is equal to the roughness under this condition. Therefore, the bounds of roughness in Section 4.6 are applicable to the degree of greyness of grey geometrical objects. It provides a convenient tool to control the uncertainty in GIS operation.

4.7.3 Application of grey geometry to GIS

In GIS, overlay operation is a routine work of most spatial queries. For simple overlay operations such as a simple overlap of each other, this does not cause any problem. However, for complicated operation, such as union, intersection and difference of two or more than two layers, the resulted map may has a completely different uncertainty associated with it. As a media of spatial information systems, it is essential to keep track with this uncertainty change and reveal its reliability to users. However, most of the present GIS do not consider this important issue at all. Some research has been carried out for this problems, and some proposals are proposed such as the buffer zone methods [9]. Here, we propose the adoption of grey geometry in dealing with this issue.

In real world measurement, a grey point represented as grey dimension $x^{\pm}, y^{\pm}$ may not be appropriate. There is no reason that the grey point can not be grey in another direction. Therefore, we propose a radius representation of grey point for GIS:

$$P(p, r) = (x, y, r)$$

Where, P is a grey point, p is a crisp point which is the centre of the grey point, and we call it core of grey point P. r is a crisp number representing the radius of the grey point, and we call it the grey span of P. In this way, a grey point is represented by a circle in a GIS map, the centre of the circle is its core, and the radius of the circle is its grey span. It is obvious that it is the relative size of the grey span of vertexes that decides the degree of greyness of a spatial object in GIS. In GIS, the most significant spatial object is polygon, so we would focus on the representation of its uncertainty in GIS. Considering the reality that most geographical measurement would have only a tiny grey span compared with the size of the segment of polygon in concern, we can approximate the degree of greyness of a polygon P as

$$g_G^{\circ}(P) = \frac{\sum_{i=1}^{n-1}[(r_i + r_{i+1}) \times l_c]}{S_c - \frac{1}{2} \times \sum_{i=1}^{n-1}[(r_i + r_{i+1}) \times l_c]}$$

Figure 4.17 demonstrates the uncertain boundary of a grey polygon. Therefore, the degree of greyness of a polygon can be easily computed using their grey vertexes. For any operation between two grey polygons, their grey vertexes would be automatically reserved together with the parts remained at the resulting polygon. If a new vertex is created between the two polygons, then the grey span will follow the one with small grey span which could be derived from a linear interpolation.

With the grey representation, a polygon in GIS will keep its uncertainty with it no matter what operation and how many operations have been carried out, and this information is essential for its final users to know its reliability after large amount of integration and overlay operations. The degree of greyness of some resulting polygon may be much larger that its original antecedents, such

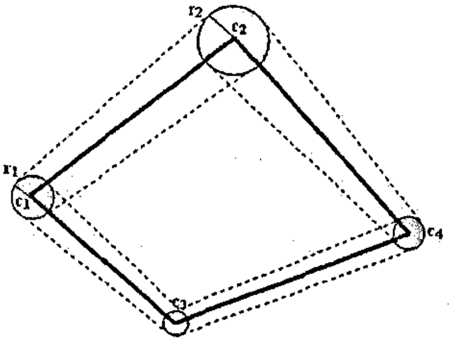


Figure 4.17: The partition problem

as a difference operations in Figure 4.18, 4.19 and 4.20. Due to the small size of a resulting polygon, its uncertainty may become too significant to its location. For example, a move of 5 metres is not a big problem for object such as a continent in the map, but it may move one house to the other side of the road.

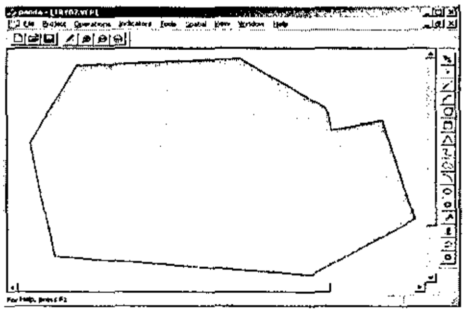

File Object Operations Indicators Tools Spatial View Window Help
For Help, press F1

Figure 4.18: Operand shape A

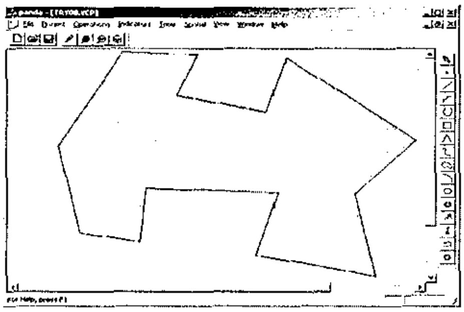

File Object Operations Indicators Tools Spatial View Window Help
For Help, press F1

Figure 4.19: Operand shape B

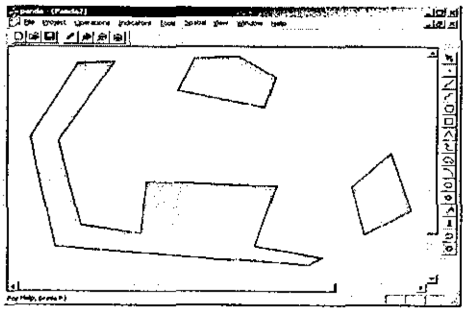


Figure 4.20: Operand shape A-B

Chapter 5

Application experiments

5.1 The role of redundant structure of neural networks

To demonstrate the performance of the redundant NN, a simple example for partition is illustrated here in Figure 5.1. Points in Figure 5.1 belong to two different parts. The input factors of the network are the coordinates of the points, and the output is 0 for points located to the left and 1 for the right. This simple problem shows the difference between the networks trained with simple inputs and redundant inputs under conditions of the same random initialisation operation. The training points include the vertex of the two parts and some inside points produced randomly.

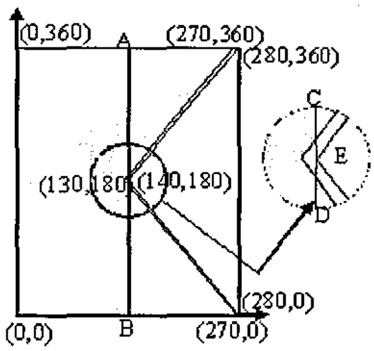


Figure 5.1: The partition problem

After some trials, the minimum requirement for the number of hidden nodes is 2 (one hidden layer). Two networks are established: each with two hidden nodes, the redundant input network with 30 sets of inputs. With exactly the same initial hidden layer and output layer as well as their connections, two networks for the same application are established under the same error limit. Their different outputs are compared in Figure 5.2. The status of points in the whole area in Figure 5.2 is calculated with the redundant input and ordinary one. Figure 5.2(a) is produced by the redundant input network, and 5.2(b) by the ordinary one. It is obvious that the redundant one gives a better resolution than the ordinary. The detail in the central area has a higher resolution. The ordinary NN is less accurate; its centre boundary is estimated as an arch which is different from Figure 5.1. The arch in the ordinary NN produces information that does not exist in the training set. This is not the best of solutions according to Jaynes' Maximum Entropy Principle [100].

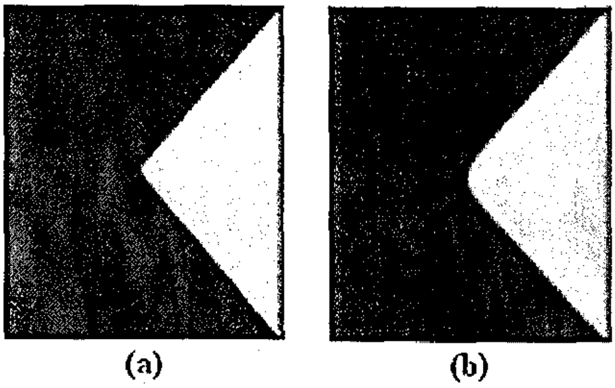


Figure 5.2: The results of compound and standard input

To test the reliability of this difference, the initial connection weights are updated with random initialization and the same experiment is repeated. In the end, all converged networks give similar differences between the two kinds of networks. However, our experiment shows also that convergence is difficult with a minimum number of hidden layer nodes: both structures have over 95% of failures (not convergent in the end). Figure 5.3 demonstrates some results from failed networks. The compound (redundant) one takes much longer. In a real world application, the optimum hidden node number is difficult to know in advance, and it is more likely that the neural network is initialised with a structure involving some extra hidden nodes. Therefore, a series of experiments for redundant hidden nodes are carried out to test their influence on neural network learning. The results demonstrate that, in addition to the improvement in the solution compared with the network with ideal network structure, a redundant structure shows also a robust feature for the possible false solution introduced by extra hidden nodes.

Figure 5.4 shows the results of this experiment for acceptable and false mapping rates which change with respect to different network size. Here, by size we

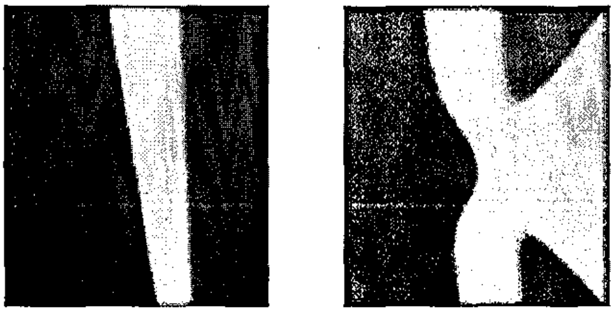
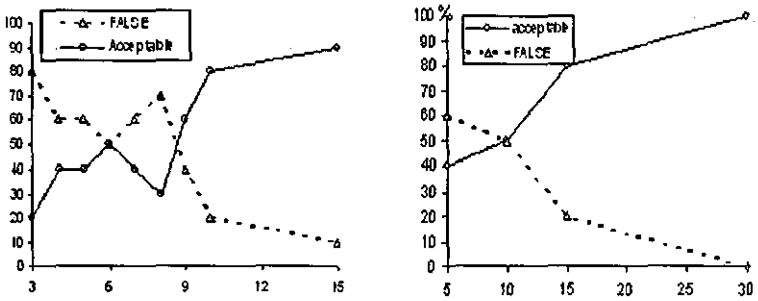


Figure 5.3: The convergent results from networks with 3 and 4 hidden nodes

mean the number of hidden nodes in the hidden layer for the ordinary network and the number of the sub input sets in redundant networks with 3 hidden nodes in the hidden layer. Figure 5.4(a) shows the result for ordinary networks and Figure 5.4(b) gives the performance of redundant input networks.



(a) (b)

Figure 5.4: The rate (%) of acceptable and false mapping vs. network size (a-ordinary network; b-redundant input)

The common perception is that a neural network of small size is more reliable. However, it is true only for the minimum size which is less likely to learn the idiosyncrasies or noise in the training data [15] - although it may involve strong non convergence as aforementioned. The example here shows that the small size with extra hidden nodes would have a high probability to give a false mapping, as demonstrated in Figure 5.4(a). This result is caused by the poor initialization of the weights. With the increment of the hidden nodes, the ideal distribution of the initalised weights approaches 0 (Equation 7) and hence the starting position for the learning is improved.

The redundant hidden nodes are sometimes inevitable for a complicated application due to our ignorance of the "black box" structure of NN. Figure 5.4(a) demonstrates that an ordinary network with 3 hidden nodes for this problem is highly likely (80%:20%) to introduce false solutions. Because of this, all the experiments in Figure 5.4(b) are based on 4 hidden nodes to test its robustness. Figure 5.4(b) shows that the increment of a redundant input network has a similar function like hidden nodes in reducing possible false solutions. However, the redundant input network has a crucial difference from the pure extra hidden nodes: it does not introduce any new parameters to the network. Hence it keeps the generality of the trained network.

Redundant structure does not always have negative effects with regard to the training of neural networks. It could play a significant role in improving the network performance under certain conditions. The redundant structure proposed here does not introduce new uncertainties into the network, but it reduces the possibility of false mappings and improves mapping quality. The method proposed is novel although simple, it does bring new problems like longer training times, but it provides a prospective direction for the improvement of neural network training operations, especially for hardware realisation.

5.2 Significance analysis using neural networks

One of the limitations of BP algorithm is the existence of local minima in its error surface. This means that it might not yield a right solution to the mapping procedure, and its RSE can not reveal the true mechanisms either. We applied two methodologies here to overcome this problem: a hybrid method of dynamic BP with random optimization [18] and the proposed redundant structure in Section 5.1. For the first method, learning begins with the modified BP method and changes into the random optimization approach when the learning process gets stuck in a local minimum. In addition, the dynamic adaptation of structure and parameters is applied to the whole process of learning where both the structure of the neural network and its learning parameters are modified dynamically according to the changes in error. In the second method, we set a relatively large redundant input sets to see if we can reduce the number of hidden units from the first method.

As mentioned above, RSE is different from the derivative of the output with respect to the input. However, if enough learning has been done, the RSE is able to approximate the scaling differentiation when the original mapping function is differentiable. Hence, the RSE method can be tested with a differentiable function for a demonstration. As an example, we consider a simple nonlinear equation as follows

$$z = 0.5x^2 + 0.1xy - 0.9y^2$$

No.	x	y	z	No.	x	y	z
1	0.37	0.81	-0.5	16	0.94	0.98	-0.33
2	0.81	0.4	0.21	17	0.06	0.49	-0.21
3	0.62	0.7	-0.2	18	0.25	0.75	-0.46
4	0.88	0.05	0.39	19	0.15	0.59	-0.3
5	0.88	0.53	0.19	20	0.59	0.79	-0.34
6	0.64	0.6	-0.08	21	0.31	0.93	-0.71
7	0.06	0.62	-0.34	22	0.41	0.29	0.02
8	0.85	0.3	0.31	23	0.81	0.13	0.32
9	0.43	0.69	-0.31	24	0.77	0.09	0.3
10	0.32	0.42	-0.1	25	0.73	0.28	0.22
11	0.85	0.76	-0.09	26	0.34	0.54	-0.19
12	0.64	0.52	-0.01	27	0.83	0.83	-0.22
13	0.56	0.97	-0.65	28	0.9	0.84	-0.16
14	0.38	0.2	0.04	29	0.42	0.46	-0.08
15	0.27	0.05	0.04	30	0.39	0.55	-0.18

Table 5.1: Training sample set of nonlinear equation

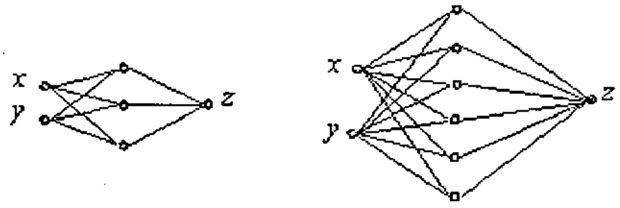
the derivative can be obtained as

$$\frac{\partial z}{\partial x} = x + 0.1y$$

$$\frac{\partial z}{\partial y} = 0.1x - 1.8y$$

We will now show that the feasibility of RSE for equation this equation can be verified by its derivative. We choose the values of x and y within [0,1] randomly, and get the corresponding z values from the given equation. One sets up a training sample set containing 30 samples as shown in Table 5.1.

The structure of a neural network is initialized as shown in Figure 5.5a. After 1044 iterations of learning, the error has been reduced to 10E-4, and the number of hidden units has been increased to 6 (Figure 5.5b) from 3. As a result of dynamic adaptation, the structure of the neural network has changed.



(a) (b)

Figure 5.5: The change of network structure for a nonlinear equation

Another group of samples which does not belong to the training set is used

No.	sample		output		
	x	y	z	z_o	error
1	0.81	0.15	0.32	0.30	-0.02
2	0.88	0.43	0.26	0.24	-0.02
3	0.63	0.37	0.1	0.09	-0.01
4	1	0.69	0.14	0.13	-0.01
5	0.56	0.32	0.08	0.08	0
6	0.23	0.97	-0.8	-0.76	0.04
7	0.28	0.18	0.01	0	-0.01
8	0.91	0.61	0.13	0.12	-0.01
9	0.63	0.36	0.1	0.1	0
10	0.49	0.74	-0.33	-0.34	-0.01

Table 5.2: The capacity of neural network for a nonlinear equation

to test the trained neural network. The testing results are listed in Table 5.2 which shows that the absolute errors between the outputs of the neural network and the ideal values are not more than 0.04. At the same time, most of their relative errors are also lower than 10% except for the 7th sample in Table 5.2. The z value of the 7th sample is too small to compare its relative error, and its absolute error is also very small.

With the trained neural network, the RSE can be obtained from Equations 4.14, and the corresponding derivatives can be worked out with the given nonlinear equation. We compared these results to test the capability of RSE as shown in Table 5.3. The first two columns are the x, y input data, and the second two columns are their derivatives. The third two columns are the derivatives divided by their maximum value, which is the control that is applied to the derivatives of the same sample. The two columns to the far right are the RSE values.

Because of the existence of error, the values of RSE are not exactly the same as the controls of the derivatives. However, it is clear enough that the values of RSE display a similar relative dominance to the controls of the derivatives shown in Table 5.3. We can make the RSE approximate the derivative with any desired degree by increasing the number of iterations. Here, we intended to demonstrate the capabilities of RSE, and we paid attention only to the relative dominance of inputs rather than the exact derivative. The comparison of RSE with the controlled derivative in Table 5.3 is shown in Figure 5.3. It is obvious that the results in Figure 5.3 show the agreement between RSE and the derivative.

For the same application, we adopt our redundant model with 30 sets of input units. However, we set 2 units in the hidden layer to check the result. After 5000 iterations of learning, we got similar error. The corresponding output test and RSE results are shown in Table 5.4 and 5.5.

Table 5.4 and 5.5 illustrate the capability of redundant network structure.

No.	input		derivative		control		RSE	
	x	y	$\partial z/\partial x$	$\partial z/\partial y$	$\partial z/\partial x$	$\partial z/\partial y$	RSE_{zx}	RSE_{zy}
1	0.2	0.2	0.22	-0.34	0.65	-1	0.58	-1
2	0.2	0.8	0.28	-1.36	0.21	-1	0.24	-1
3	0.8	0.2	0.82	-0.28	1	-0.34	1	-0.41
4	0.8	0.8	0.88	-1.28	0.61	-1	0.63	-1
5	0.5	0.5	0.55	-0.85	0.65	-1	0.69	-1
6	0.5	0.2	0.52	-0.31	1	-0.6	1	-0.7
7	0.2	0.5	0.25	-0.88	0.28	-1	0.32	-1
8	0.5	0.8	0.58	-1.31	0.44	-1	0.39	-1
9	0.8	0.5	0.85	-0.82	1	-0.96	1	-0.83

Table 5.3: The derivative and RSE of a nonlinear equation

No.	sample		output		
	x	y	z	z_o	error
1	0.81	0.15	0.32	0.32	0
2	0.88	0.43	0.26	0.24	-0.02
3	0.63	0.37	0.1	0.09	-0.01
4	1	0.69	0.14	0.14	0
5	0.56	0.32	0.08	0.07	-0.01
6	0.23	0.97	-0.8	-0.76	0.04
7	0.28	0.18	0.01	0	-0.01
8	0.91	0.61	0.13	0.13	0
9	0.63	0.36	0.1	0.09	-0.01
10	0.49	0.74	-0.33	-0.35	0.02

Table 5.4: The capacity of the redundant network for a nonlinear equation

No.	input		derivative		control		RSE	
	x	y	$\partial z/\partial x$	$\partial z/\partial y$	$\partial z/\partial x$	$\partial z/\partial y$	RSE_{zx}	RSE_{zy}
1	0.2	0.2	0.22	-0.34	0.65	-1	0.68	-1
2	0.2	0.8	0.28	-1.36	0.21	-1	0.29	-1
3	0.8	0.2	0.82	-0.28	1	-0.34	1	-0.60
4	0.8	0.8	0.88	-1.28	0.61	-1	0.69	-1
5	0.5	0.5	0.55	-0.85	0.65	-1	0.61	-1
6	0.5	0.2	0.52	-0.31	1	-0.6	1	-0.73
7	0.2	0.5	0.25	-0.88	0.28	-1	0.29	-1
8	0.5	0.8	0.58	-1.31	0.44	-1	0.37	-1
9	0.8	0.5	0.85	-0.82	1	-0.96	1	-0.91

Table 5.5: The derivative and RSE of a nonlinear equation from redundant network

Although there is a corresponding traditional network with the same capability, but it is very difficult to find in most cases, and our redundant structure provides an efficient way to reduce hidden layer units so as to reduce the extra parameters. The reduction of those extra parameters help to reduce the complexity and hence increase the reliability of trained networks.

This nonlinear equation is simple, but the result has demonstrated the efficiency of RSE. Clearly, the RSE reflects the dynamic variation of the effect for inputs acting on output. With the same operation, one can analyze more complicated problems.

5.3 A new method to evaluate a trained artificial neural network

We have shown in last section that an ill-defined neural network will give false results. Hence it is necessary to evaluate a trained neural network to see if it gives a correct mapping. In Section 4.3, we have proposed a novel method in evaluating a trained neural network. To illustrate the applicability of the proposed evaluation method, we use the same partition example as illustrated in Figure 5.1. Obviously, the maximum significance of the two dimensions (x,y) are exactly the same: no influence to the output in the left side of AB and same importance along the boundary interface. Therefore, the field knowledge advises:

$$GPRSE_x = GPRSE_y$$

$$PRSE_x^a = PRSE_y^a$$

where, 'a' represents an arbitrary point in the two parts of Figure 5.1.

For a simple problem like this example, it is applicable to test the trained neural network in the whole scope of interest. Therefore, the traditional testing method is applied here to validate the applicability of the new approach proposed here. As suggested in Section 4.3, a three layer network (one hidden layer) has the ability to approach any continuous mapping. Hence, we investigate only the structure with one hidden layer here. The structure with 3,4,5,6,7,8,9,10 and 15 hidden nodes are studied with a random initialised connection weights within [0,1]. For every structure, 10 converged networks with different initial connection weights are established.

Coincidently, the acceptable and false mappings are equivalent in the 90 established networks: 45 for true and the other 45 for false. The true mapping appears similar with each other, as shown in Figure 5.6. However, the false mapping shows the diversity of the possible false solutions, two examples are demonstrated in Figure 5.7. It should be noted that all the false mappings in these experiments converge well and have the potential to converge further.

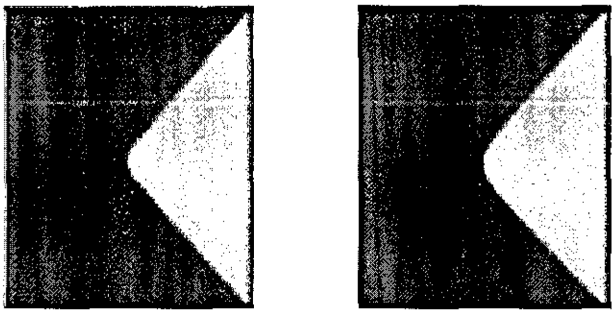


Figure 5.6: The results of two acceptable mappings

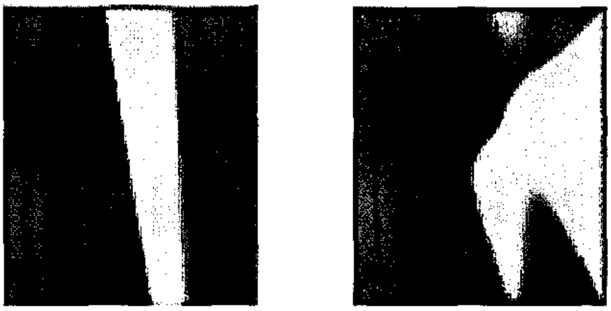


Figure 5.7: Some results of the false mappings

Obviously, the results of the acceptable mapping and the false mapping are very different. It could be easily found for a simple problem by this exhaustive testing method. However, it is not so easy for a complex engineering problem where a large scale network involved. Because of the potential complexity of the engineering problems, it is impossible to test every possible situation in fact. The only way for testing it with the traditional method is to keep large amount of samples out of the training set and then test the trained network with this reserved data set. As aforementioned, this operation reduces the available data set for training the network and can not prove the applicability of the network for the data not included in the testing set in the end. For example, the false mappings in Figure 5.7 may find a lot of satisfactory test results if the test points are not located in the distorted areas. Hence, the conclusions based on this kind of local sample testing are not reliable if we cannot make an efficient distribution of the test data.

Based on the trained neural networks, the GPRSE results for the acceptable mapping and false mapping are calculated and illustrated in Figure 5.8.

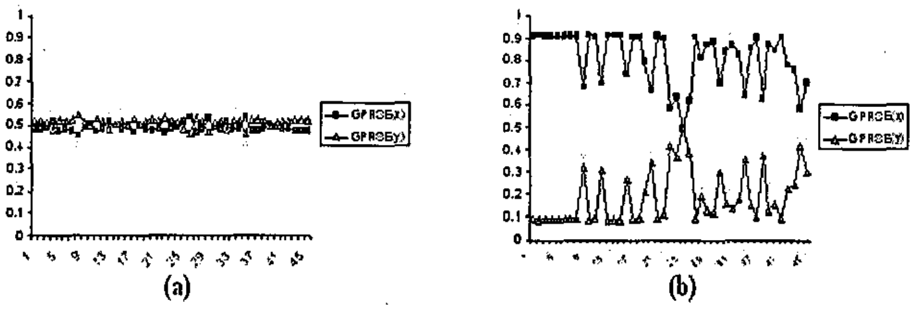


Figure 5.8: The comparison of GPRSE

The Figure 5.8(a) shows the GPRSE for the acceptable group, and Figure 5.8(b) is the GPRSE for the false mapping group. The GPRSE pairs for x and y distributed along the line of GPRSE=0.5, it proves that the GPRSEx is close to GPRSEy for the acceptable group. On the contrary, most GPRSE pairs for the false mapping group are far from each other in Figure 5.8(b). Obviously, the false mapping is reflected well by their GPRSE, that is: GPRSEx GPRSEy for the false mapping. Compared with aforementioned field knowledge, it is reasonable to evaluate the trained network by means of GPRSE: the network would be false mapping if its GPRSE pairs for x and y have a large distance from each other. Among the 90 trained networks, only one is an exception to this rule, and the probability for this rule to identify the false mapping is larger than 0.99.

The exception of the pure GPRSE identification is in the false mapping group. As demonstrated in Figure 5.8(b), the network 26 (6 hidden nodes) possesses a GPRSE pair of 0.499 for x and 0.501 for y. According to the first equation in aforementioned field knowledge, this network should be an acceptable one. However, its exhaustive testing shows a distorted result as shown in Figure 5.9.

Obviously, this mapping forms another kind of pattern with equivalent GPRSE pairs for this problem. Hence the GPRSE identification does not work in this special case. However, although this pattern keeps the equivalent GPRSE pairs, but its local distortion could be found with some local PRSE evaluation. For example, we select one centre point in every partition to do the PRSE analysis. Here, the points are (70,180) for the left partition and (210,180) for the right part. The result of their mapping from the exceptional network 26 in the false mapping group and a normal network 11 in the acceptable group are listed in Table 5.6. The column 'O_i' means the ideal output, and 'O_n' is the real output from the trained network.

The two points selected here located in the centre of the two parts and the

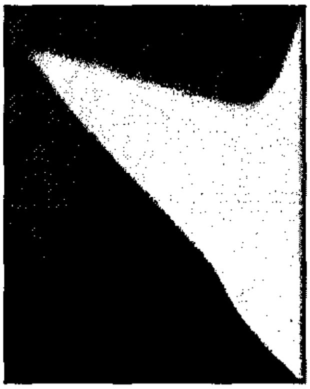


Figure 5.9: The result of the exception case

Group	No.	x	y	O_i	O_n	$PRSE_x$	$PRSE_y$
FALSE	26	70	180	0	0	0.43	0.57
FALSE	26	210	180	1	1	0.71	0.29
Accept	11	70	180	0	0	0.48	0.52
Accept	11	210	180	1	1	0.48	0.52

Table 5.6: Comparison between the two cases

variance of x or y would not causes the change of the output within their neighborhood. Therefore, their PRSE should be equivalent to each other. Obviously, the network 11 in the acceptable group agrees with this analysis well, but the exceptional case in the false mapping group does not meet this requirement. However, the mapping results of both cases are satisfactory. Obviously, the traditional testing method fails to find the problem in this case, and the proposed PRSE analysis works well.

Generally, the GPRSE identification works well if the general trends is clear enough. For complicated problems, the general trends may not be so clear and then PRSE may be involved to validate it further.

An exhaustive testing for the possible situations for a neural network by means of the testing samples is impossible and not necessary. It would reduce the limited available data for training the neural networks in engineering practice. Hence this work presents a new methodology to combine the field knowledge on the trends analysis with the network validation process [208].

As a global parameter, the GPRSE is a very good indicator for the behaviour of a trained neural network. As the index of the importance of the input parameters on determining the output, GPRSE should agree with the special field knowledge of the relative role of the individual input parameter. For a complicated problem where GPRSE is not clear enough, the PRSE analysis serves for further investigation. Therefore, an analysis of the specific field knowledge

INDEX	OUTLOOK	TEMP(F)	HUMIDITY(%)	WINDY?	CLASS
1	sunny	75	70	true	Play
2	sunny	80	90	true	Don't Play
3	sunny	85	85	false	Don't Play
4	sunny	72	95	false	Don't Play
5	sunny	69	70	false	Play
6	overcast	72	90	true	Play
7	overcast	83	78	false	Play
8	overcast	64	65	true	Play
9	overcast	81	75	false	Play
10	rain	71	80	true	Don't Play
11	rain	65	70	true	Don't Play
12	rain	75	80	false	Play
13	rain	68	80	false	Play
14	rain	70	96	false	Play

Table 5.7: Training data set for data mining [142]

combining with a few sample tests would give a better audit to the efficiency of the trained artificial neural networks. In this way, the available data could be fully applied to the training stage and the validation is simple and efficient.

5.4 Data mining using neural networks

For the sake of simplicity, a very small data set on weather conditions and play decisions is used to demonstrate the utility of this new approach and shown in Table 5.7 [142]. "Outlook" represents the day with or without cloud and rain. An "outlook" can be clear sky without cloud or heavy rain with full cloud. Hence, cloudiness is a condition for rain and represents the degree of the possibility of rain.

Bearing these in mind, we regard "outlook" as a fuzzy concept about the weather with two ultimate states: sunny or rain, and overcast as a middle state. The fuzzy membership for outlook is thus considered as a linear function. Similarly, "Windy?" and "Class" items are considered as fuzzy concepts with two ultimate states: true or false for "windy" and play or don't play for "class". A similar linear membership is also assigned to them. To increase the number of the available data set, we insert middle values into the intervals of the original data set and calculate the output as a linear value. However, only the original data set is adopted as the candidate explanation points in the DSS operation.

A neural network consisting of 3 layers is established. The RSE values for the training points are listed in Table 5.8.

From the values in this Table it is obvious that the RSE values are dynamic

Index	Outlook	Temperature	Humidity	Windy
1	-0.66	-0.588	-1	0.171
2	-1	-0.644	-0.682	0.365
3	-0.761	-0.425	-1	-0.408
4	-1	-0.503	-0.721	-0.215
5	-1	-0.578	-0.877	-0.006
6	-1	-0.878	-0.752	-0.078
7	-1	-0.554	-0.897	-0.1
8	1	0.128	0.087	-0.519
9	-0.996	-0.58	-1	-0.133
10	1	0.174	0.292	-0.847
11	0.502	-0.053	-0.646	-1
12	0.709	-0.017	-0.164	-1
13	1	0.123	0.046	-0.919
14	-1	-0.563	-0.494	-0.641

Table 5.8: The RSE values for the training set

Corresponding Original Concepts	Values after Transition	Group
Outlook: Raining	0	Input
Temperature: 75	0.5 (60-90)	
Humidity: 70	0.14 (65-100)	
Windy: True	1	
Class: Don't Play	0.0456	Output
RSE for Outlook	0.764	
RSE for Temperature	0.047	
RSE for Humidity	-0.326	
RSE for Windy	-1	

Table 5.9: The input and output of the new case

from one sample to the others. When a new case is fed into the NN, a new group of RSE values could be calculated with the network in the same way. These serve as the dimensions of significance in DSS as shown in Figure 4.12. This significance is a dynamic index because it changes with the variance of the different combinations of weather conditions shown in Table 5.8. In this way, our new approach is able to trace the dynamic changes in the different roles of the same parameter in different situations.

As an example, a new case (Outlook: Raining, Temperature: 75, Humidity: 70, Windy: True) is fed into the trained NN. The input, output and RSE of the network are shown in Table 5.9.

Comparing Table 5.8 with Table 5.9, it is clear that the new case produces a new group of RSE values which are different from those in Table 5.8. Therefore, the dominant role is sensitive to the different combinations of the weather conditions here. Hence a DSS approach is more likely to give a reasonable conclusion compared with a traditional space approach.

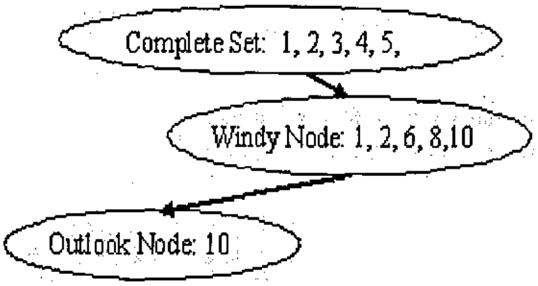


Figure 5.10: Search in DSS

Following the process shown in Figure 4.12, the DSS search strategy is established as shown in Figure 5.10. The "Windy" item has the highest absolute value of RSE (1.0), so it should be the first attribute to be searched in the DSS as aforementioned. The DSS focuses only on those cases near to the active cases, hence only the sample 1, 2, 6, 8, 10 and 11 are left after the first dimension search (Dimension path). Because the relevant set has more than one case and there are still available dimensions left, the search needs to be extended to the next dimension. In addition to "Windy", "Outlook" has a second big absolute value of RSE, and it is applied as the second attribute here to be investigated for a further search of the DSS. It is obvious that only samples 10 and 11 are left after the suitable neighbours search because all others in the "windy" node have a different "outlook" when compared with the active case. The extension of the DDP can be stopped at this point because "temperature" and "humidity" have very low RSE absolute values here and the two samples in the suitable set have output similar to the output of the active case from the trained network. Therefore, a simple DSS as demonstrated in Figure 5.10 is established.

The explanation for the result of this new case can be derived from the DSS search shown in Figure 5.10. For this case, the two relevant cases 10 and 11 are similar to the new case. Their two most significant coordinates are exactly the same, therefore their output would be the same.

The explanation can be described as follows:

The most important attributes are "Windy" (RSE=1) and "Outlook" (RSE=0.764) for the new case. There are two very similar cases in the training samples: 10 and 11, and both have the same output "Don't play". Therefore, the new case should have the same output.

The lack of a readily understandable explanation for the 'black box' operation of neural networks prohibits some potential further deployment. The current emphasis on rule extraction from neural networks is difficult to marry up with the powerful capability of neural networks in mapping complicated relationships. Based on the Relative Strength of Effect (RSE) approach, a new type of neural network explanation strategy has been presented in this paper - based on Dynamic State Space search and RSE explanations. Unlike current neural network rule extraction methods, our approach does not attempt to extract all possible knowledge present in a system; rather, it aims to extract an explanation which is understandable only in the context of a specific query about the mapping results of a neural network.

As an illustration, a small data set on the relationship between weather conditions and play decisions was presented to demonstrate the utility of the proposed approach. The example showed that the DSS search from RSE was an effective alternative in the resulting explanation of the NN. It traced those significant weather conditions in a dynamic way and provided a meaningful explanation of the output of the NN.

5.5 Noise evaluation using neural networks

Noise disturbance at airports is the most significant environment problem at the moment, and most airports are adopting noise models for their operation planning. As aforementioned in Section 2.3, most existing noise models are based on some standard tuned data sets. For example, INM adopts NDP data sets as its foundation [41]. Normal NDP data consist of two or more noise curves [4]. A noise curve reflects the relationship between distances and noise levels under specific engine power (thrust in pounds) and operation mode (departure or approach) under a standard condition. However, these curves give only measurements at the following 10 distances: 200, 400, 630, 1000, 2000, 4000, 10000, 16000 and 25000 feet. Any noise level in between these measurements or in between those given thrusts has to be evaluated using mathematical models, such as linear interpolation, logarithmic interpolation and extrapolation. However, these mathematical models are established against a standard measuring environment at a specific airport for the test. The geographical condition and environment parameters at other airports may not be the same as the testing airport, so models established in INM may not give results as near to the real world measurement as expected. To adjust those parameters in INM to suit the local geographic and environmental condition at an airport is complicated and difficult, and there are many mathematical models involved into these processes to consider the relationship between noise level and temperature, wind speed and direction, and other acoustics factors. Due to the complexity of natural environment at an airport, these models cannot fit with every airport and are bound to produce further errors and uncertainty. Therefore, a simple way of establishing noise simulation at a local airport would be a great help in airport noise simulation and operation planning. Here, we adopt neural networks as the universal models for adapting standard NDP curves to local conditions.

5.5.1 Available data

Because of the significant impact of aircraft noise on airport development, most large airports in the world have already started to monitor the noise level in the vicinity of airports. With the incorporation of Manchester airport in our EPSRC research "A decision support system for sustainable airport development", we collected some monitored noise records from two monitor stations: Kell House Farm and Broad Oak Farm. Their locations are shown in Figure 5.11.

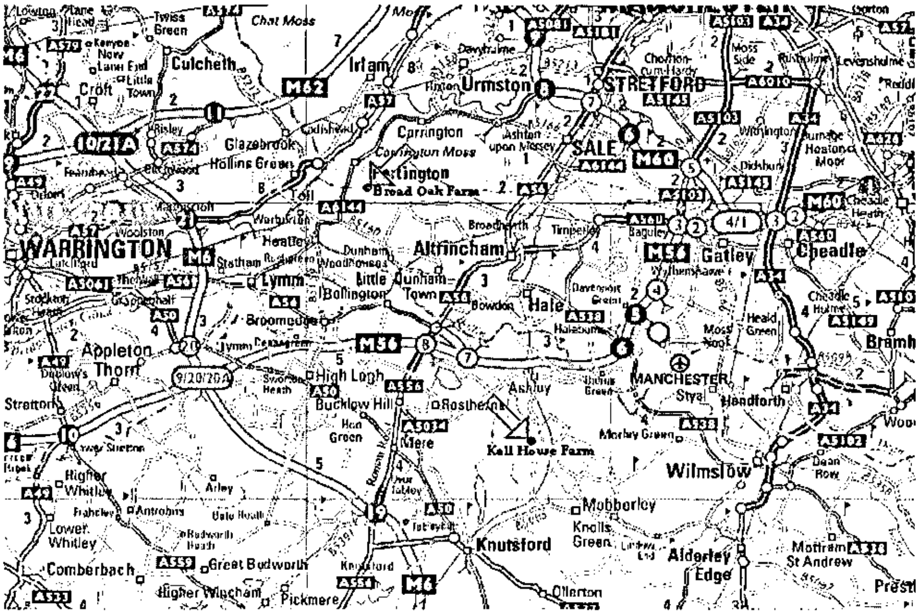


Figure 5.11: Location of the two monitor stations

The data were monitored during the period of 1998–2001, and the largest volume of data are recorded for B757: 10408 records from Kell House Farm and 7368 records from Broad Oak Farm. The recorded data attributes include aircraft type, operation mode, direct distance to the monitoring station, station name, maximum noise level recorded and its recording time. Considering the volume of the data, we show it by its noise distribution against distance under departure or approach operation in Figure 5.12, 5.13, 5.14 and 5.15.

The monitored data in Figure 5.12, 5.13, 5.14 and 5.15 scatter everywhere along the same distance. Obviously, a general model suitable to each airport is very difficult to establish for such data set without knowing more information

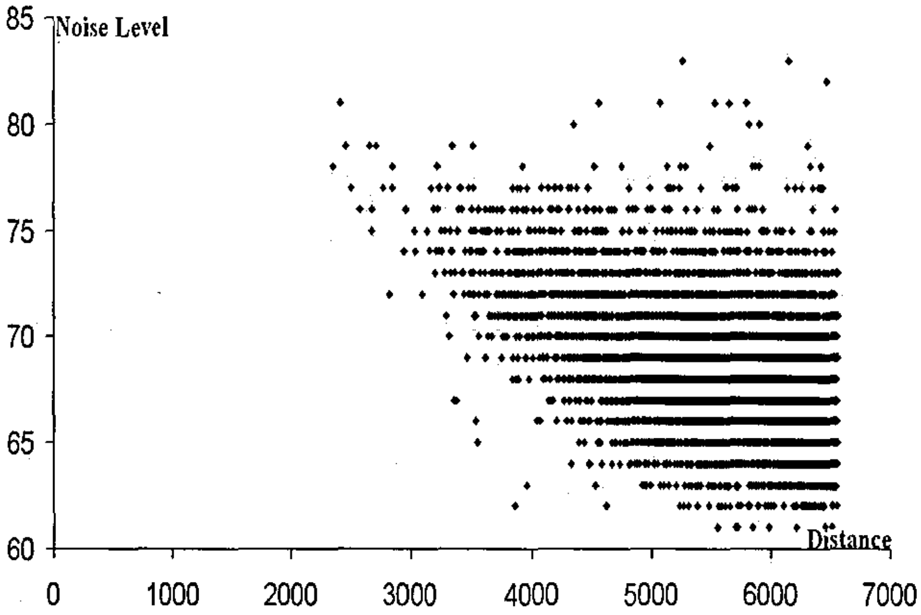


Figure 5.12: Noise distribution against distance for departing flights recorded at Kell House Farm station

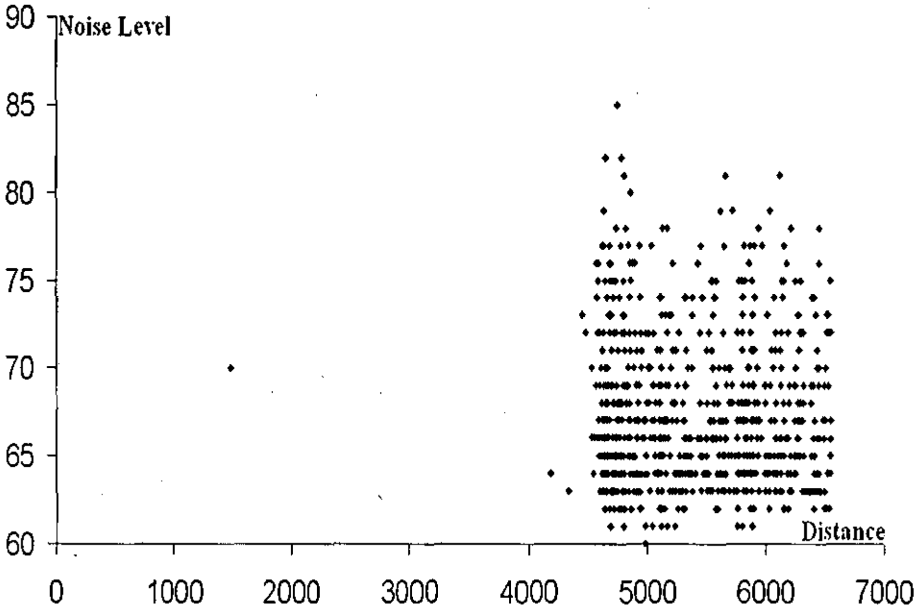


Figure 5.13: Noise distribution against distance for approaching flights recorded at Kell House Farm station

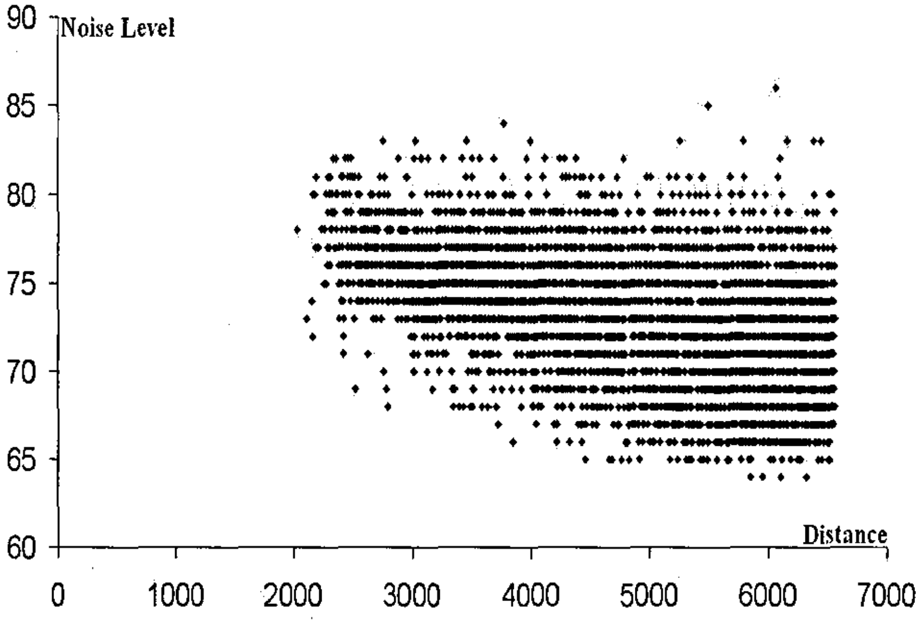


Figure 5.14: Noise distribution against distance for departing flights over Broad Oak Farm station

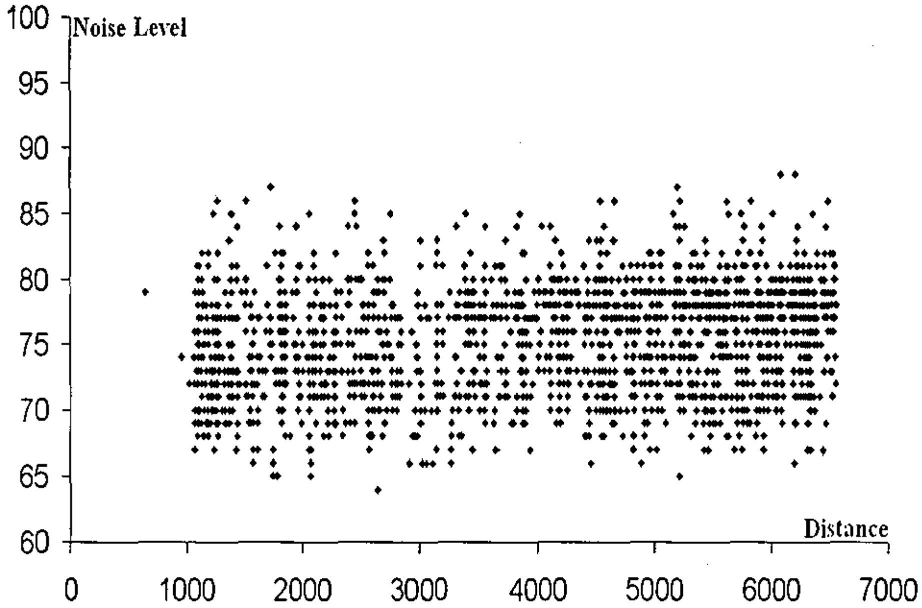


Figure 5.15: Noise distribution against distance for approaching flights over Broad Oak Farm station

about the flight speed, thrust, weight, trajectory, wind speed and direction, terrain of the airport etc. Among these factors, the weather conditions and geographical features of the vicinity of an airport would not be as different as those between two airports. The monitored data are measured at the local weather and geographical condition, hence their influence to the noise level at the relevant airport has already been embedded into the monitored data. A model established from the monitored data is suitable only to the airport where data are collected. Therefore, the weather and geographical conditions are not as significant as distances and thrusts of flights for a local noise model. Therefore, NDP curves are accurate as long as the weather and geographical conditions at a local airport match those conditions of a standard testing airport. The disagreement between INM model results and in-situ monitoring data comes from the difference between their weather and geographical conditions. Therefore, a local NDP curve could be established considering the same relationships as INM: the relationship among noise levels, operation modes, distances and thrust. Other factors are not significant for the observations in the vicinity of the same airport. The data in Figure 5.12, 5.13, 5.14 and 5.15 have attributes for noise levels, distances and operation mode. However, thrust is missing in the collected data set. Actually, for the same distance at the airport and observation location, there is more than one point in all 4 figures. It is mainly caused by their different thrust at that distance. Because of the automatic landing control for the direction of approach, the thrust is often changed during approach operation and causes larger fluctuation of the points in Figure 5.13 and 5.15 than 5.12 and 5.14. It proves that thrust is a significant factor determining the monitored noise levels. Therefore, it is essential to get the thrusts of those monitored points in Figure 5.12, 5.13, 5.14 and 5.15. However, the thrusts are not recorded in the monitored data, and it is very difficult to get it due to the large number of possible flight trajectories.

5.5.2 Reverse map thrust using neural networks and NDP data

We made an attempt to establish neural networks without thrusts, but the result is not satisfactory. The result is especially poor when a network trained using data from one monitoring station is applied to the other station. They are not better than an average estimation in most cases. It proves again that thrusts are essential in the noise evaluation in airports. Now that NDP curves provide the relationships among noise levels, operation modes, distances and thrust, it is possible to establish a model to do a reverse map to find thrust from known noise level as well. Neural networks provide ideal tools in doing this reverse mapping from the available standard NDP curves. There will be difference between the obtained thrust and those measured in flights. However, it is possible to establish mathematical models to adjust an obtained thrust into a measured thrust. Compared with adjustment of noise at every concerned location involving huge amount of factors, it is much easy to adjust a single thrust from a single aircraft engine.

Based on the aforementioned idea, we need to establish the reverse map from known noise levels to their corresponding thrusts. Before establishing the reverse map, we need to evaluate the capacity of neural networks in mapping the NDP curves. We adopt the data from standard NDP database in INM here. The NDP databases in INM contains a set of NDP data for 224 aircraft types. There are four kinds of NDP noise data:

- L_{AE} A-weighted sound exposure level;
- L_{ASmx} Maximum A-weighted sound level with slow-scale exponential time weighting;
- L_{EPN} Effective tone-corrected perceived noise level;
- L_{PNTSmx} Maximum tone-corrected perceived noise level with slow-scale exponential time weighting.

The aim of the first experiment is to investigate the capability of neural networks in simulating NDP curves. To this end, we adopt the aircraft with the maximum number of available data in NDP databases. After comparing data in the database using SQL, B737-200 has the maximum number of rows in the database. Therefore, we adopt B737-200 data as the test bed for neural network simulation. To test the trained neural networks, we separate the original NDP data into two different groups: those data measured at a distance of 200, 630, 2000, 6300 and 16000 feet are used as training data to establish the neural network. The data measured at 400, 1000, 4000, 10000 and 25000 feet are applied as testing data. The two data sets are shown in Table 5.10 and 5.11. In these two tables, the column "L_200" refers the noise levels at a distance of 200 feet. Same explanation applies to other columns starting with "L". "ACFT_DESCR" indicates the aircraft type, "NOISE_TYPE" represents the 4 aforementioned noise types, "OP_MODE" differentiates the two operation mode approach (A) and departure (D), and "THR_SET" is the thrust value.

Using the training data in Table 5.10, a neural network with 10 sets of compound inputs and 8 nodes in the hidden layer is established. The inputs include noise type, operation mode, distance and thrust, and the output is noise level. After 5000 iterations, the error is reduced to below 2E-4. With the

ACFT_DESCR	NOISE_TYPE	OP_MODE	THR_SET	L_200	L_630	L_2000	L_6300	L_16000
B737-200	S	D	16000.00	115.90	109.50	100.80	90.10	79.80
B737-200	S	D	14000.00	113.10	106.70	98.00	87.30	77.00
B737-200	S	D	12000.00	110.20	103.50	94.70	83.90	73.70
B737-200	S	D	10000.00	106.60	100.00	91.30	80.70	70.60
B737-200	S	D	8000.00	102.70	96.10	87.60	77.20	67.40
B737-200	S	D	6000.00	98.60	92.00	83.40	73.00	63.10
B737-200	S	A	5000.00	98.50	92.10	83.00	71.40	60.00
B737-200	S	A	3000.00	94.00	87.50	78.40	66.70	55.30
B737-200	P	D	16000.00	125.80	114.50	101.40	86.70	73.40
B737-200	P	D	14000.00	123.50	112.20	99.10	84.30	71.00
B737-200	P	D	12000.00	120.90	109.50	96.20	81.20	67.80
B737-200	P	D	10000.00	117.20	106.10	92.80	77.80	64.10
B737-200	P	D	8000.00	113.40	102.40	89.20	74.20	60.50
B737-200	P	D	6000.00	109.30	98.00	84.70	69.50	55.70
B737-200	P	A	5000.00	110.40	97.10	82.20	65.90	51.60
B737-200	P	A	3000.00	105.80	92.70	77.70	60.90	46.00
B737-200	M	D	16000.00	113.80	103.40	90.90	76.50	63.30
B737-200	M	D	14000.00	111.40	101.00	88.50	74.10	60.90
B737-200	M	D	12000.00	108.50	98.10	85.50	71.00	57.80
B737-200	M	D	10000.00	105.10	94.70	82.30	67.90	54.80
B737-200	M	D	8000.00	101.20	90.90	78.50	64.40	51.60
B737-200	M	D	6000.00	96.80	86.50	74.10	59.90	47.00
B737-200	M	A	5000.00	95.80	84.80	72.40	58.80	46.60
B737-200	M	A	3000.00	90.60	79.70	67.30	53.60	41.50
B737-200	E	D	16000.00	117.40	111.00	102.70	92.70	83.20
B737-200	E	D	14000.00	114.50	108.20	99.90	89.80	80.30
B737-200	E	D	12000.00	111.30	104.90	96.60	86.50	76.90
B737-200	E	D	10000.00	108.40	101.60	93.00	82.50	72.70
B737-200	E	D	8000.00	104.60	97.70	88.90	78.50	68.60
B737-200	E	D	6000.00	100.60	93.60	84.40	74.10	64.10
B737-200	E	A	5000.00	103.30	93.80	82.70	70.30	59.10
B737-200	E	A	3000.00	99.70	90.30	79.00	66.00	54.20

Table 5.10: The training NDP data set for B737-200

ACFT_DESCR	NOISE_TYPE	OP_MODE	THR_SET	L_400	L_1000	L_4000	L_10000	L_25000
B737-200	S	D	16000.00	112.30	106.30	94.60	85.20	74.40
B737-200	S	D	14000.00	109.50	103.50	91.80	82.40	71.60
B737-200	S	D	12000.00	106.40	100.20	88.40	79.00	68.30
B737-200	S	D	10000.00	102.90	96.80	85.10	75.90	65.30
B737-200	S	D	8000.00	99.00	92.90	81.50	72.50	62.30
B737-200	S	D	6000.00	94.80	88.80	77.40	68.30	57.90
B737-200	S	A	5000.00	94.90	88.80	76.30	66.00	53.90
B737-200	S	A	3000.00	90.40	84.20	71.70	61.30	49.30
B737-200	P	D	16000.00	110.20	109.50	92.70	80.20	66.60
B737-200	P	D	14000.00	116.90	107.20	90.40	77.90	64.30
B737-200	P	D	12000.00	114.20	104.40	87.40	74.70	60.90
B737-200	P	D	10000.00	110.70	101.00	84.00	71.10	57.00
B737-200	P	D	8000.00	107.00	97.40	80.40	67.60	53.50
B737-200	P	D	6000.00	102.70	92.90	75.70	62.80	48.70
B737-200	P	A	5000.00	102.50	91.30	72.50	58.90	44.40
B737-200	P	A	3000.00	98.10	86.90	67.70	53.60	38.40
B737-200	M	D	16000.00	107.80	98.70	82.50	70.10	56.60
B737-200	M	D	14000.00	105.40	96.30	80.10	67.80	54.20
B737-200	M	D	12000.00	102.50	93.30	77.00	64.60	51.00
B737-200	M	D	10000.00	99.10	90.00	73.80	61.60	48.00
B737-200	M	D	8000.00	95.20	86.10	70.20	58.20	45.00
B737-200	M	D	6000.00	90.80	81.80	65.80	53.70	40.40
B737-200	M	A	5000.00	89.30	80.00	64.30	52.90	40.50
B737-200	M	A	3000.00	84.20	74.90	59.20	47.70	35.40
B737-200	E	D	16000.00	113.70	107.90	96.90	88.10	78.20
B737-200	E	D	14000.00	110.90	105.10	94.00	85.30	75.30
B737-200	E	D	12000.00	107.70	101.80	90.70	81.90	71.90
B737-200	E	D	10000.00	104.50	98.40	86.90	77.80	67.60
B737-200	E	D	8000.00	100.70	94.40	82.80	73.70	63.50
B737-200	E	D	6000.00	96.60	90.30	78.50	69.30	58.80
B737-200	E	A	5000.00	97.70	89.50	75.40	64.90	53.30
B737-200	E	A	3000.00	94.20	86.00	71.40	60.30	48.10

Table 5.11: The testing NDP data set for B737-200

established neural networks, we get the noise prediction as shown in Table 5.12 for those test data in Table 5.11.

Obviously, the trained neural networks has obtained a reasonable mapping capability in the NDP curves, and all test results are above 90% in accuracy. The percentage of records with a noise level lower than a given difference or error is demonstrated in Table 5.13 and Figure 5.16.

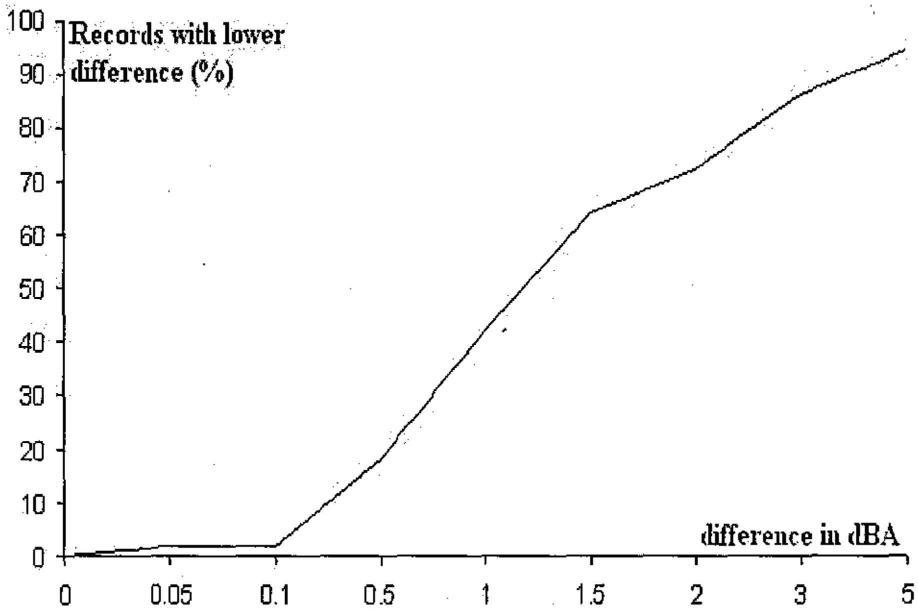


Figure 5.16: Noise prediction against B737 test data (NDP)

From Table 5.13 and Figure 5.16, it is clear that the prediction accuracy increases with the control difference. It demonstrates that an acceptable prediction is about 2 dBA in difference. This is actually caused by our big distance span in the data set. We hold half of data for test, and use only half the data to train the network, which results in the large distance span in the data set. For example, "L_400" is taken out as test data, this leaves no data between "L_200" and "L_630". Therefore, the prediction accuracy would not be as high as if we use the whole data set as training data. However, our purpose here is to prove the capability of the neural network mapping for NDP curves. To this end, the trained neural network has been provided with enough evidence.

Having verified the capability of neural networks in mapping NDP curves, we now establish a neural network to do reverse mapping to get the missing thrusts for our in-situ data. Here, to match our monitored data, we adopt only the maximum A-weighted sound level with slow-scale exponential time weighting L_{ASmx}. The aim of this experiment is to simulate NDP curves using neural networks so as to get those missing thrust values in our data set. For this

ID	NOISE_TYPE	OP_MODE	THR_SET	Distance	Noise_Level	ANN Output	Error	Accuracy
1	E	D	12000	400	107.7	108.819	1.119	98.96
2	E	D	14000	400	110.9	111.8634	0.9634	99.13
3	E	A	5000	400	97.7	97.8967	0.1967	99.79
4	E	D	6000	400	96.6	97.1337	0.5337	99.45
5	E	D	10000	400	104.5	105.354	0.854	99.18
6	E	D	16000	400	113.7	114.5012	0.8012	99.29
7	E	A	3000	400	94.2	93.2741	-0.9259	99.02
8	E	D	8000	400	100.7	101.4599	0.7599	99.24
9	M	A	5000	400	89.3	89.9427	0.6427	99.28
10	M	D	6000	400	90.8	90.2614	-0.5386	99.41
11	M	A	3000	400	84.2	84.8067	0.6067	99.28
12	M	D	8000	400	95.2	94.7469	-0.4531	99.52
13	M	D	16000	400	107.8	108.8854	1.0854	98.99
14	M	D	10000	400	99.1	98.8235	-0.2765	99.72
15	M	D	12000	400	102.5	102.5172	0.0172	99.98
16	M	D	14000	400	105.4	105.8613	0.4613	99.56
17	P	D	6000	400	102.7	102.0657	-0.6343	99.38
18	P	D	8000	400	107	105.9786	-1.0214	99.05
19	P	D	10000	400	110.7	109.2891	-1.4109	98.73
20	P	D	12000	400	114.2	112.0714	-2.1286	98.14
21	P	A	3000	400	98.1	96.1784	-1.9216	98.041
22	P	D	16000	400	119.2	116.371	-2.829	97.63
23	P	D	14000	400	116.9	114.407	-2.493	97.87
24	P	A	5000	400	102.5	101.0659	-1.4341	98.61
25	S	A	5000	400	94.9	95.9135	1.0135	98.93
26	S	D	6000	400	94.8	95.115	0.315	99.67
27	S	D	8000	400	99	99.6523	0.6523	99.34
28	S	D	10000	400	102.9	103.6343	0.7343	99.29
29	S	D	12000	400	106.4	107.0807	0.6807	99.36
30	S	A	3000	400	90.4	90.5793	0.1793	99.80
31	S	D	16000	400	112.3	112.596	0.296	99.74
32	S	D	14000	400	109.5	110.046	0.546	99.50
33	E	D	12000	1000	101.8	102.218	0.418	99.59
34	E	D	14000	1000	105.1	105.5612	0.4612	99.56
35	E	A	5000	1000	89.5	90.1519	0.6519	99.27
36	E	D	6000	1000	90.3	90.2744	-0.0256	99.97
37	E	D	10000	1000	98.4	98.5548	0.1548	99.84
38	E	D	16000	1000	107.9	108.5846	0.6846	99.36
39	E	A	3000	1000	86	85.6667	-0.3333	99.61
40	E	D	8000	1000	94.4	94.5724	0.1724	99.82
41	M	A	5000	1000	80	81.5967	1.5967	98.00
42	M	D	6000	1000	81.8	82.6205	0.8205	98.99
43	M	A	3000	1000	74.9	76.8372	1.9372	97.41
44	M	D	8000	1000	86.1	86.9159	0.8159	99.05
45	M	D	16000	1000	98.7	101.3155	2.6155	97.35
46	M	D	10000	1000	90	90.9094	0.9094	98.99
47	M	D	12000	1000	93.3	94.6197	1.3197	98.58
48	M	D	14000	1000	96.3	98.0783	1.7783	98.15
49	P	D	6000	1000	92.9	95.7413	2.8413	96.94
50	P	D	8000	1000	97.4	99.8504	2.4504	97.48
51	P	D	10000	1000	101	103.4285	2.4285	97.59
52	P	D	12000	1000	104.4	106.5221	2.1221	97.97
53	P	A	3000	1000	86.9	88.956	2.056	97.63
54	P	D	16000	1000	109.5	111.5108	2.0108	98.16
55	P	D	14000	1000	107.2	109.1945	1.9945	98.14
56	P	A	5000	1000	91.3	93.9266	2.6266	97.12
57	S	A	5000	1000	88.8	89.8343	1.0343	98.84
58	S	D	6000	1000	88.8	89.575	0.775	99.13
59	S	D	8000	1000	92.9	94.0372	1.1372	98.78
60	S	D	10000	1000	96.8	98.036	1.236	98.72
61	S	D	12000	1000	100.2	101.5693	1.3693	98.64
62	S	A	3000	1000	84.2	84.6765	0.4765	99.43
63	S	D	16000	1000	106.3	107.4119	1.1119	98.96
64	S	D	14000	1000	103.5	104.6757	1.1757	98.86
65	E	D	12000	4000	90.7	88.3741	-2.3259	97.44
66	E	D	14000	4000	94	01.2756	-2.7244	97.10
67	E	A	5000	4000	75.4	74.5656	-0.8344	98.89
68	E	D	6000	4000	78.5	77.7667	-0.7333	99.07
69	E	D	10000	4000	86.9	85.158	-1.742	97.99
70	E	D	16000	4000	96.9	93.8799	-3.0201	96.88
71	E	A	3000	4000	71.4	70.7051	-0.6949	99.03
72	E	D	8000	4000	82.8	81.6195	-1.1805	98.57
73	M	A	5000	4000	64.3	62.9174	-1.3826	97.85
74	M	D	6000	4000	65.8	65.4328	-0.3672	99.44
75	M	A	3000	4000	59.2	59.351	0.151	99.74
76	M	D	8000	4000	70.2	69.0926	-1.1074	98.42
77	M	D	16000	4000	82.5	82.1048	-0.3952	99.52
78	M	D	10000	4000	73.8	72.6081	-1.1919	98.38
79	M	D	12000	4000	77	75.9489	-1.0511	98.64
80	M	D	14000	4000	80.1	79.1107	-0.9893	98.76
81	P	D	6000	4000	75.7	78.0105	2.3105	96.95
82	P	D	8000	4000	80.4	81.8727	1.4727	98.17
83	P	D	10000	4000	84	85.4164	1.4164	98.31
84	P	D	12000	4000	87.4	88.6226	1.2226	98.60
85	P	A	3000	4000	67.7	69.5857	1.8857	97.21
86	P	D	16000	4000	92.7	94.1067	1.4067	98.48
87	P	D	14000	4000	90.4	91.5074	1.1074	98.78
88	P	A	5000	4000	72.5	73.7197	1.2197	98.32
89	S	A	5000	4000	76.3	75.6447	-0.6553	99.14
90	S	D	6000	4000	77.4	77.0096	-0.3904	99.49

Table 5.12: The test results for B737

Difference(dBA)	Records with higher difference	Records with lower difference (%)
0	160	0
0.05	157	1.875
0.1	157	1.875
0.5	131	18.125
1	93	41.875
1.5	57	64.375
2	44	72.5
3	22	86.25
5	9	94.375

Table 5.13: Noise level testing result for B737 testing data

experiment, we need the same aircraft as the one with the maximum number of available monitored data. Here, the aircraft determined by our monitored data is B757. Therefore, we adopt the NDP data for B757. The data are shown in Table 5.14.

The NDP data for each aircraft are very limited and we have to fully make use of the available data. Here, the "leave-out-one cross validation" method is adopted in the training of neural networks. From the experiment in Section 5.1, redundant structure could be applied if compound inputs were established. Here, we adopt a compound input of 10 sets and set the hidden layer node number as 6. The inputs are operation mode (OP_MODE), maximum noise level (Noise_Level) and distance, the output is the thrust. After 30000 iterations using "leave-out-one cross validation", the errors are reduced to lower than 2.0E-4. The cross validation results are shown in Table 5.15.

In Table 5.15, "Output" is the result from the trained neural network. The values in the two columns are close to each other in all rows, and their maximum difference is less than 14%, and over 88% rows have difference less than 5%. Table 5.16 gives the number of records with errors lower than the given error in the first column (%) and their corresponding accuracy. The result is also demonstrated in Figure 5.17.

It proves that the trained neural network is valid to derive thrusts from the measured noise levels. From the trained neural network, we can get the GRSE and GPRSE as shown in Table 5.17.

The distance has the dominant role in determining thrust, and the noise has important role as well. The operation seems not very significant to thrust. The conclusions agree with our data set: a longer distance and lower noise level indicate less thrust from the engine. The operation mode determines if the thrust is stable or not, but it does not determine its values.

Using the trained neural networks, we got the missing thrusts for the moni-

OP_MODE	THR_SET	Noise_Level	Distance
D	24000.00	101.50	200
A	5000.00	95.10	200
D	13000.00	97.40	200
D	30000.00	103.30	200
D	36000.00	105.50	200
A	12000.00	99.60	200
D	24000.00	97.20	400
A	5000.00	90.80	400
D	13000.00	93.30	400
D	30000.00	99.60	400
D	36000.00	102.50	400
A	12000.00	95.00	400
D	24000.00	94.00	630
A	5000.00	87.70	630
D	13000.00	90.20	630
D	30000.00	96.70	630
D	36000.00	100.20	630
A	12000.00	91.70	630
D	24000.00	90.40	1000
A	5000.00	84.30	1000
D	13000.00	86.80	1000
D	30000.00	93.60	1000
D	36000.00	97.70	1000
A	12000.00	88.10	1000
D	24000.00	84.70	2000
A	5000.00	78.70	2000
D	13000.00	80.90	2000
D	30000.00	88.30	2000
D	36000.00	93.30	2000
A	12000.00	82.10	2000
D	24000.00	77.90	4000
A	5000.00	72.00	4000
D	13000.00	73.90	4000
D	30000.00	81.90	4000
D	36000.00	87.30	4000
A	12000.00	75.20	4000
D	24000.00	73.20	6300
A	5000.00	66.80	6300
D	13000.00	68.60	6300
D	30000.00	77.10	6300
D	36000.00	82.30	6300
A	12000.00	69.90	6300
D	24000.00	68.20	10000
A	5000.00	61.10	10000
D	13000.00	62.50	10000
D	30000.00	71.90	10000
D	36000.00	76.50	10000
A	12000.00	63.90	10000
D	24000.00	62.70	16000
A	5000.00	54.70	16000
D	13000.00	55.80	16000
D	30000.00	66.00	16000
D	36000.00	69.70	16000
A	12000.00	57.20	16000
D	24000.00	57.00	25000
A	5000.00	48.20	25000
D	13000.00	48.80	25000
D	30000.00	59.90	25000
D	36000.00	62.60	25000
A	12000.00	50.40	25000

Table 5.14: The NDP data set for B757

ID	OP_MODE	Noise_Level	Distance	THR_SET	Output	Error(%)	Accuracy(%)
1	D	101.5	200	24000	26596.23	10.82	89.18
2	A	95.1	200	5000	5568.96	11.38	88.62
3	D	97.4	200	13000	14714.59	13.19	86.81
4	D	103.3	200	30000	31375.67	4.59	95.41
5	D	105.5	200	36000	36216.89	0.60	99.40
6	A	99.6	200	12000	13054.69	8.79	91.21
7	D	97.2	400	24000	23287.58	2.97	97.03
8	A	90.8	400	5000	4756.88	4.86	95.14
9	D	93.3	400	13000	12667.77	2.56	97.44
10	D	99.6	400	30000	29489.32	1.70	98.30
11	D	102.5	400	36000	35607.06	1.09	98.91
12	A	95	400	12000	11287.57	5.94	94.06
13	D	94	630	24000	22941.51	4.41	95.59
14	A	87.7	630	5000	4844.48	3.11	96.89
15	D	90.2	630	13000	12774.53	1.73	98.27
16	D	96.7	630	30000	29430.99	1.90	98.10
17	D	100.2	630	36000	35950.00	0.14	99.86
18	A	91.7	630	12000	11640.76	2.99	97.01
19	D	90.4	1000	24000	23224.82	3.23	96.77
20	A	84.3	1000	5000	5109.48	2.19	97.81
21	D	86.8	1000	13000	13537.38	4.13	95.87
22	D	93.6	1000	30000	30240.69	0.80	99.20
23	D	97.7	1000	36000	36493.77	1.37	98.63
24	A	88.1	1000	12000	12443.16	3.69	96.31
25	D	84.7	2000	24000	23174.27	3.44	96.56
26	A	78.7	2000	5000	4811.40	3.77	96.23
27	D	80.9	2000	13000	12802.22	1.52	98.48
28	D	88.3	2000	30000	30378.65	1.26	98.74
29	D	93.3	2000	36000	36076.92	0.21	99.79
30	A	82.1	2000	12000	11790.73	1.74	98.26
31	D	77.9	4000	24000	23226.52	3.22	96.78
32	A	72	4000	5000	4677.88	6.44	93.56
33	D	73.9	4000	13000	12613.89	2.97	97.03
34	D	81.9	4000	30000	30660.87	2.20	97.80
35	D	87.3	4000	36000	35473.90	1.46	98.54
36	A	75.2	4000	12000	11653.14	2.89	97.11
37	D	73.2	6300	24000	23997.99	0.01	99.99
38	A	66.8	6300	5000	5144.88	2.90	97.10
39	D	68.6	6300	13000	13461.06	3.55	96.45
40	D	77.1	6300	30000	30851.30	2.84	97.16
41	D	82.3	6300	36000	35625.38	1.04	98.96
42	A	69.9	6300	12000	12323.32	2.69	97.31
43	D	68.2	10000	24000	23272.40	3.03	96.97
44	A	61.1	10000	5000	5172.64	3.45	96.55
45	D	62.5	10000	13000	12854.63	1.12	98.88
46	D	71.9	10000	30000	30115.38	0.38	99.62
47	D	76.5	10000	36000	35765.91	0.65	99.35
48	A	63.9	10000	12000	11831.20	1.41	98.59
49	D	62.7	16000	24000	23677.24	1.34	98.66
50	A	54.7	16000	5000	4893.96	2.12	97.88
51	D	55.8	16000	13000	13122.47	0.94	99.06
52	D	66	16000	30000	30124.92	0.42	99.58
53	D	69.7	16000	36000	36237.75	0.66	99.34
54	A	57.2	16000	12000	11744.92	2.13	97.87
55	D	57	25000	24000	24197.14	0.82	99.18
56	A	48.2	25000	5000	5205.95	4.12	95.88
57	D	48.8	25000	13000	12798.09	1.55	98.45
58	D	59.9	25000	30000	30183.23	0.61	99.39
59	D	62.6	25000	36000	35705.06	0.82	99.18
60	A	50.4	25000	12000	11895.75	0.87	99.13

Table 5.15: The test results for a neural network established from NDP data of B757

Error (%)	Records with lower error	Accuracy (%)
0	60	0.00
1	46	23.33
2	32	46.67
3	20	66.67
4	11	81.67
5	6	90.00
6	5	91.67
7	4	93.33
8	4	93.33
9	3	95.00
10	3	95.00
15	0	100.00

Table 5.16: Mapping results of the reverse neural network from NDP data

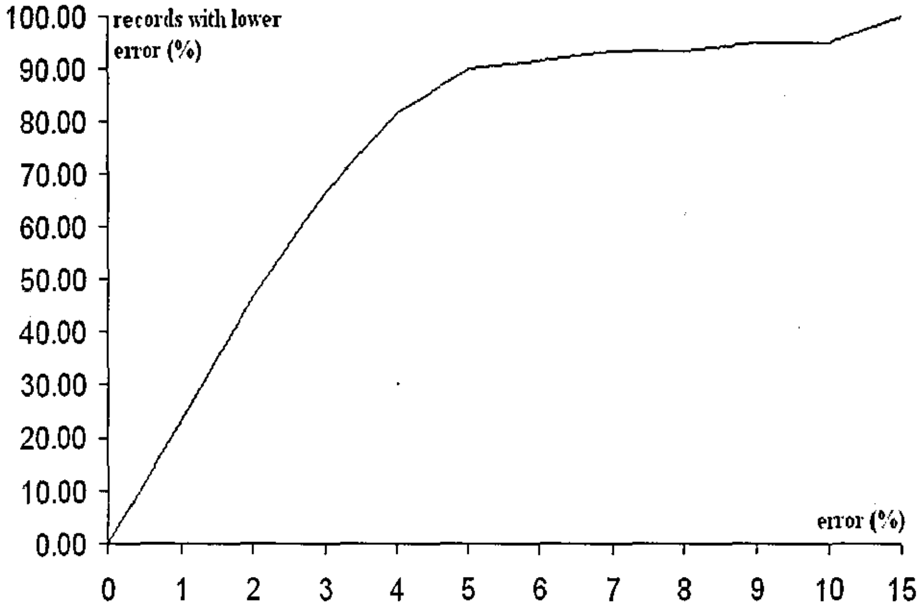


Figure 5.17: Mapping results of the revers mapping neural network from NDP data

Factor	GRSE	GPRSE
Operation mode	-0.02	0.03
Noise level	0.23	0.20
Distance	1.0	0.78

Table 5.17: GRSE and GPRSE for a neural network established from NDP data of B757

tored data. As suggested by GRSE, the distance is the dominant factor of the thrust values, so we demonstrate their distribution against distance in Figure 5.18, 5.19, 5.20 and 5.21.

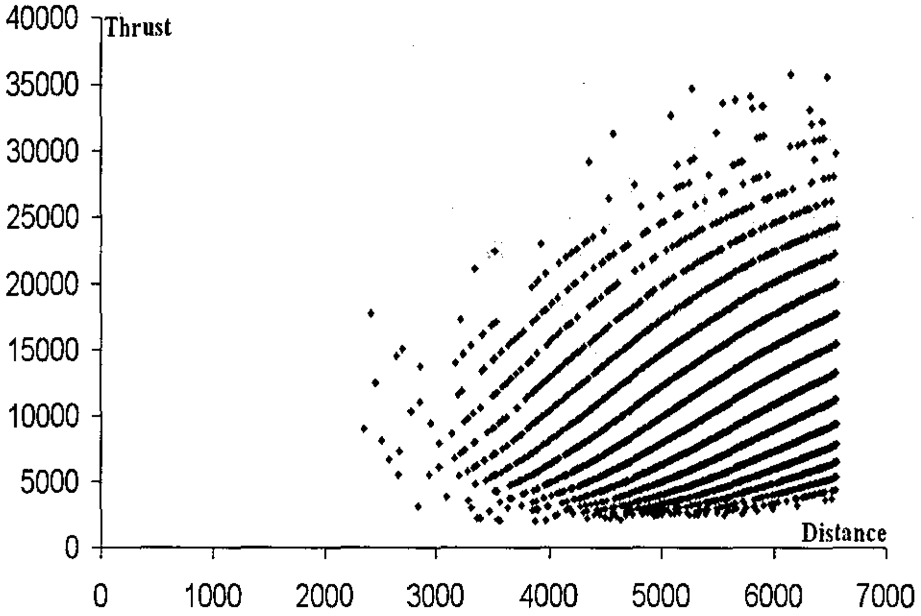


Figure 5.18: Thrust distribution against distance for departing flights over Kell House Farm station

Data in Figure 5.18, 5.19, 5.20 and 5.21 demonstrate very high relevance to data in Figure 5.12, 5.13, 5.14 and 5.15. The noise level values in Figure 5.12, 5.13, 5.14 and 5.15 show some line patterns, and these results in the curve patterns in Figure 5.18, 5.19, 5.20 and 5.21. For the same noise level, the thrust values increase with distances. For the same distance, thrust increases with noise values and jumps from one curve to a curve above it. Obviously, it is the same as that has been revealed by GRSE and GPRSE. It proves the efficiency of GRSE and GPRSE.

5.5.3 Noise level prediction using neural networks

Having got the thrusts for each record, we establish a neural network using the measured data at Kell House Farm station. Similar to the NDP network, we adopt the same input parameters: distance between an aircraft and the station, operation mode of the aircraft and thrust of the aircraft. The output is the maximum noise level at the monitoring station. For the sake of speed, we use 10 sets of inputs again in the compound structure. We use 10 sets of input as compound inputs again, and 8 hidden layer nodes. There are 10408 records

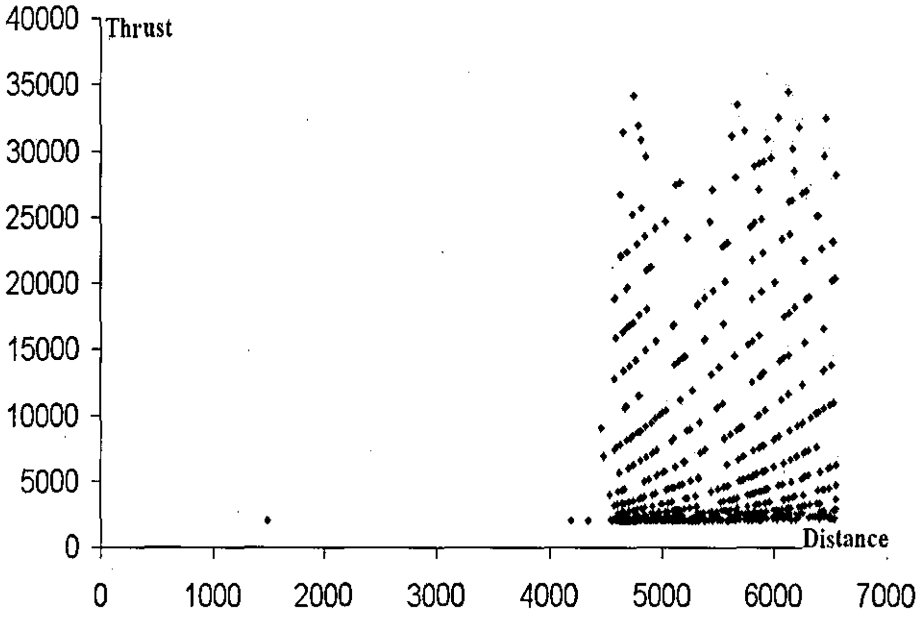


Figure 5.19: Thrust distribution against distance for approaching flights over Kell House Farm station

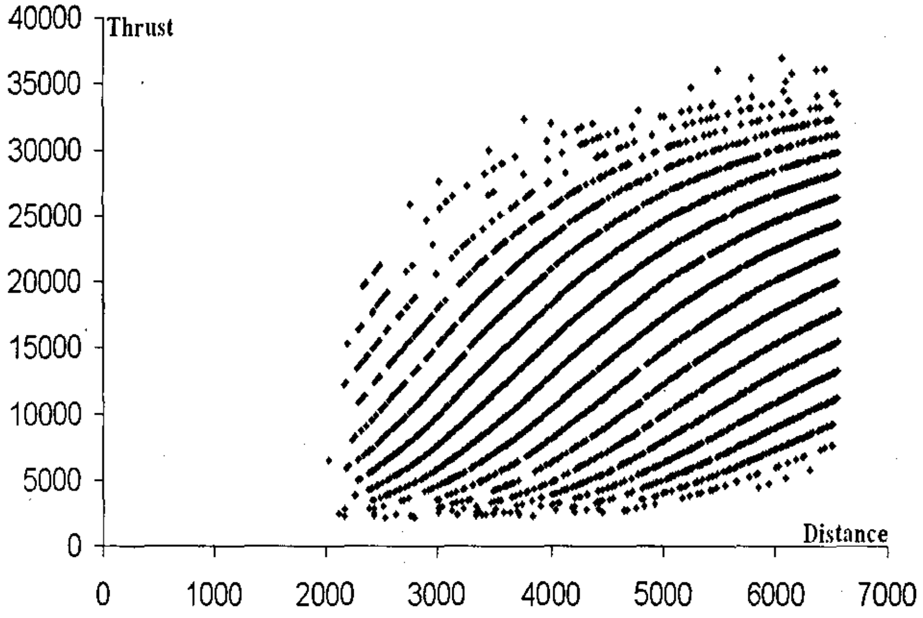


Figure 5.20: Thrust distribution against distance for departing flights over Broad Oak Farm station

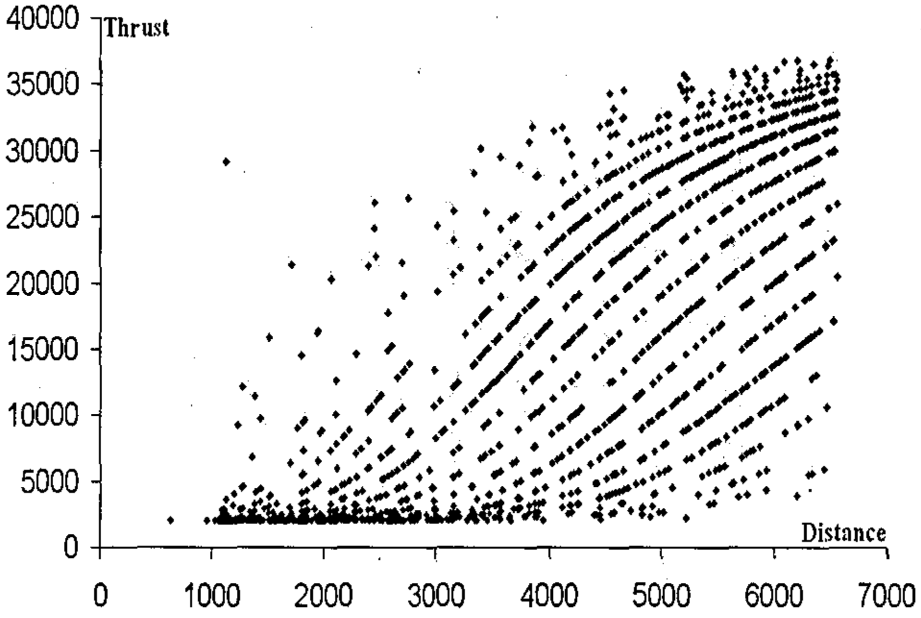


Figure 5.21: Thrust distribution against distance for approaching flights over Broad Oak Farm station

for B757 at Kell House Farm station. We separate the data into two different groups: each record with an odd number as index is hold as training data, and each row with an even number as index is kept as testing data. In this way, we have 5204 rows in both groups. After 5000 iterations, the error is reduced to lower than 1.0e-4. The testing result is shown in Table 5.18 and Figure 5.22.

In Table 5.18 and Figure 5.22, the "difference" refers to the difference between the measured noise level and the output noise level from the trained neural network. The "records with lower difference (%)" represents the percentage of records with a noise level lower than the corresponding difference. It

Difference(dBA)	Records with higher difference	Records with lower difference (%)
0	4208	0
0.05	3251	37.53
0.1	1879	63.89
0.5	160	96.92
1	65	98.75
1.5	18	99.65
2	9	99.82
3	3	99.94
5	1	99.98

Table 5.18: Noise level testing result for Kell House Farm station

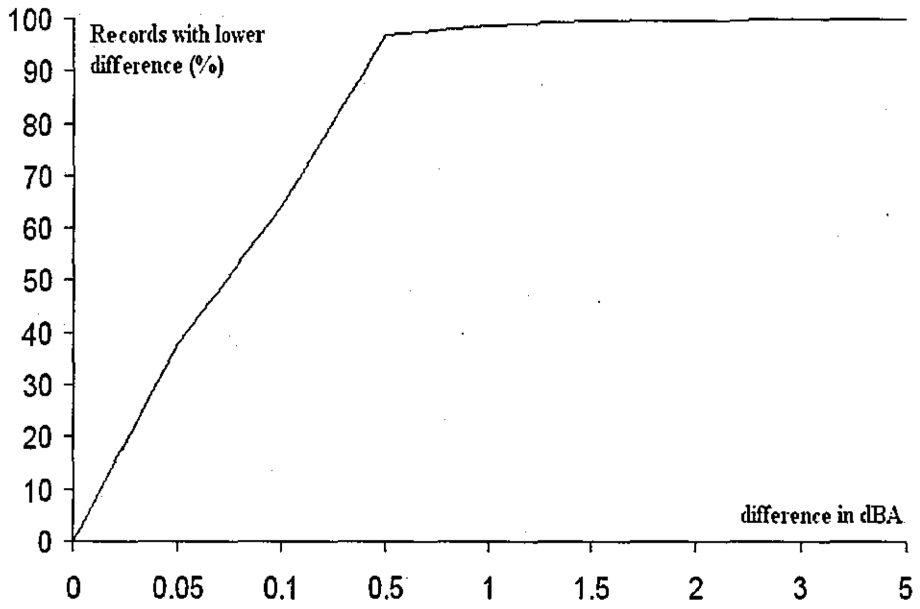


Figure 5.22: Noise prediction against data from Kell House Farm station

is obvious that the percentage of records with lower difference is very high for difference over 0.5 dBA. It is a very accurate prediction for noise level at airports. However, this result is a test for the data measured at the same station, and it needs a further test against data from a different station. Therefore, we carried out the test using data measured at Broad Oak Farm station. There are 7368 records measured at Broad Oak Farm station. We applied the trained NDP neural network to get the thrust, then using the obtained thrusts together with measured distances, operation modes to derive the noise levels using our trained neural networks form Kell House Farm data. The result is demonstrated in Table 5.19 and Figure 5.23.

Comparing Figure 5.22 and 5.23, the prediction accuracy for Broad Oak Farm station is slightly lower than the test results from test data at Kell House Farm station. However, considering the model is established with data from Kell House Farm station, the test results at Broad Oak Farm station are actually acceptable. For prediction difference of 0.5 dBA, there are nearly 85% of the predictions are accurate enough. Obviously, with the involvement of thrusts, it is possible to give a very high accuracy in the prediction of noise level at airports.

One would wonder if NDP model is good enough in doing this. To test the capacity of a neural network trained from NDP data, we established a similar model using NDP data in Table 5.14. The structure of the network is exactly the same as the network for Kell House Farm station. Using "leave one out"

Difference(dBA)	Records with higher difference	Records with lower difference (%)
0	7363	0.07
0.05	5472	25.73
0.1	4308	41.53
0.5	1111	84.92
1	651	91.16
1.5	547	92.58
2	485	93.42
3	320	95.66
5	87	98.82

Table 5.19: Noise level testing result for Broad Oak Farm station

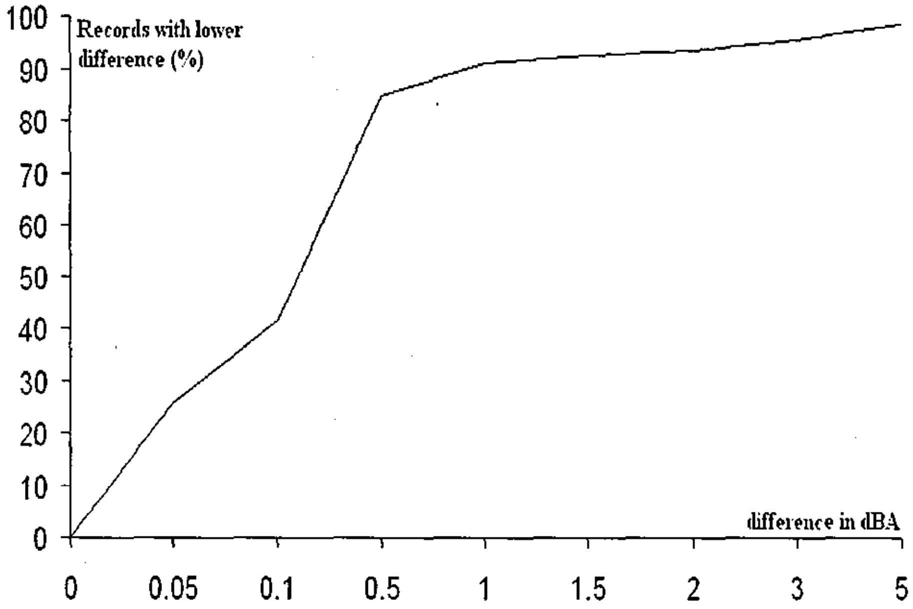


Figure 5.23: Noise prediction against data from Broad Oak Farm station

Difference(dBA)	Records with higher difference	Records with lower difference (%)
0	5204	0
0.05	5008	3.77
0.1	4794	7.88
0.5	3129	39.87
1	1379	73.50
1.5	638	87.74
2	301	94.22
3	93	98.21
5	16	99.69

Table 5.20: Noise level testing result for Kell House Farm station using NDP network

cross validation method, we established the NDP neural network for noise level. Applying this NDP neural network, we got the test results for the test data at Kell House Farm station as shown in Table 5.20 and Figure 5.24.

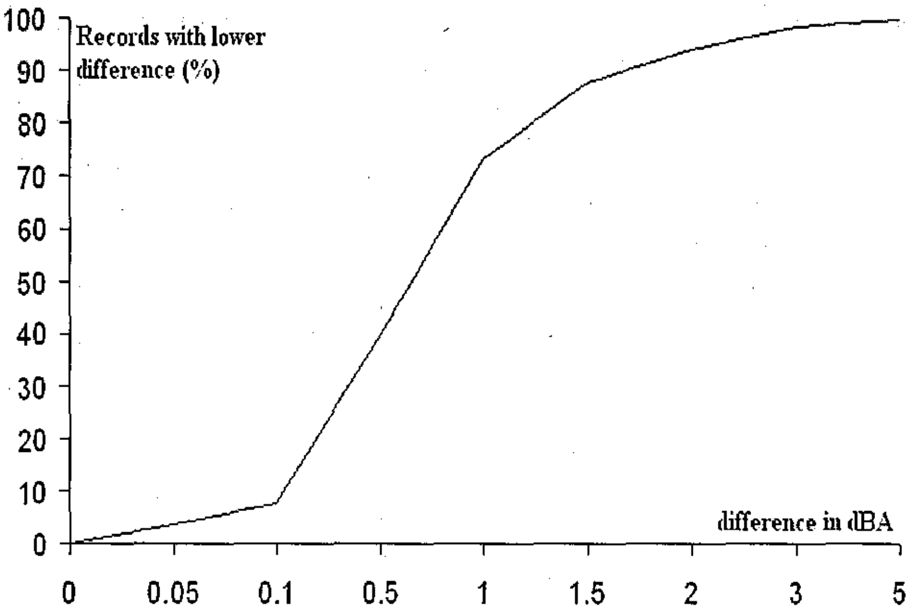


Figure 5.24: Noise prediction against data from Kell House Farm station using NDP network

Comparing Figure 5.22, 5.23 and 5.24, it is clear that the neural network trained with NDP data could give reasonable prediction (70%) only when the difference is at 1 dBA or above, and its prediction is very poor for 0.5 dBA (40%). However, the neural network trained with data measured at Kell House station could give much better results for both data sets. To check the prediction

Difference(dBA)	Records with higher difference	Records with lower difference (%)
0	7168	0
0.05	7165	2.76
0.1	6951	5.66
0.5	4990	32.27
1	2066	71.96
1.5	1339	81.83
2	1201	83.70
3	1011	86.28
5	703	90.46

Table 5.21: Noise level testing result for Broad Oak Farm station using NDP network

quality of the neural network trained with NDP data for data measured at the Broad Oak Farm station, we applied the trained network of NDP data to our data set from Broad Oak Farm station. The results are given in Table 5.21 and Figure 5.25.

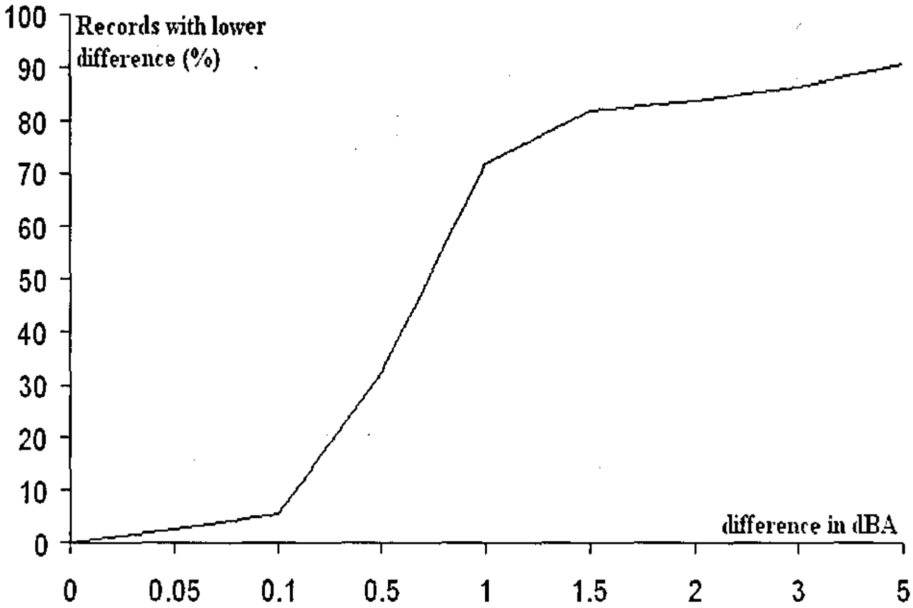


Figure 5.25: Noise prediction against data from Broad Oak Farm station using NDP network

The results in Table 5.21 and Figure 5.25 are very similar to the results in Table 5.20 and Figure 5.24. It proves that the model obtained from NDP data gives similar result to both sets of data. The poor results from the model trained with NDP data are caused by the difference between the geographi-

Input	NDP		Kell House Farm	
	GRSE	GPRSE	GRSE	GPRSE
Operation mode	-0.014	0.049	0.012	0.085
Distance	-1	0.772	-0.064	0.1
Thrust	0.171	0.179	1	0.815

Table 5.22: GRSE and GPRSE of the trained neural networks using NDP data and Kell House Farm data

cal and weather conditions of the standard airport condition of NDP test and Manchester airport. This complex relationship is very difficult to tune with INM, but it is very convenient using neural networks. The data are measured at Manchester airport, so its values have already reflected the geographical and weather conditions at Manchester airport. The trained model is not applicable to other airport, but has better performance at the local airport.

The GRSE and GPRSE values of the two models trained with NDP data and Kell House Farm data are listed in Table 5.22.

The data in Table 5.22 show that the dominant factor in NDP model is distance, but it changes to thrust in the model trained with Kell House Farm data. The NDP data are obtained under standard airport under the condition that the aircraft is flying parallel to the sea surface. However, the monitored data come from various operations of aircraft when they take off or land into the airport under difference complicated weather conditions. Therefore, the frequency of the change of their thrust is much higher than the standard testing conditions. This means that noise level is more frequently influenced by thrust rather than distance in the real operation condition. Therefore, the model from NDP data is bound to give high errors for the real operation. It provides evidence in the other hand that airports could reduce noise level by improved operation under the same geographical conditions.

5.5.4 Probability model

Although our model does give satisfactory results, it is not reliable to give accurate noise level prediction due to the very fact that an accurate thrust is difficult to get. In addition to thrust, there are also other factors difficult to quantify under real operation of the airport, such as the dynamic wind speed and direction etc. Therefore, under the same condition of the distance, operation and thrust, there are different noise levels existing in the data sets. This causes some very significant errors for some individual test records. This kind of errors would never be able to be removed from the neural network models considering only limited factors. Therefore, a more reasonable way in evaluating airport noise is the evaluation of its probability to get noise over some given noise level. Bear this in mind, we processed the data sets to create noise probability distribution data. According to INM model, noise level has a linear relationship with thrust and a logarithmic relationship with distance [4]. It is reasonable to consider the probability of noise over some given noise level following the same relationships. Therefore, we divide the thrust and distance into intervals and derive the probability of each point using its two adjacent intervals, as shown in Figure 5.26.

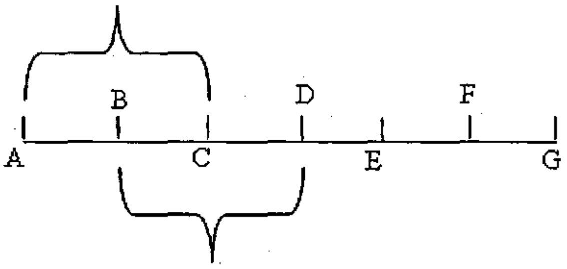


Figure 5.26: The probability calculation method

In Figure 5.26, B, C, D, E and F are the location points dividing AB into equal intervals in distance or thrust. To derive the probability of noise level over a given noise level L, we use those measured data located within the two adjacent intervals. For example, if we want to get the probability for noise level over 70 dBAs at location B, we can count all the measured noise levels located within the interval AB and BC and calculate the probability P_B as

$$P_B = \frac{\text{number of records with noise level higher than 70 within AC}}{\text{number of all records within AC}}$$

Because of the linear relationship, we can have larger intervals for thrust, and the distance interval would need to be smaller because of the proposed logarithmic relationship in INM. In this way, we can make better use of the available data.

Following this idea, we divide the thrust into 8 intervals, and distance into 50 intervals. The probability is calculated as aforementioned. Here, we control the minimum number of records required in each location as 3 records to take account those irregular records. The noise levels are also divided into 20 different levels as the reference noise. Therefore, for each given noise level, we have 7 × 50 probability points, and we have 7000 points in total. We calculate the probability for each operation mode separately, and hence we would have 14000 points if all of them had more than 3 records in their adjacent intervals.

With the B757 data sets at Kell House Farm station, we got 7180 rows in

Difference(dBA)	Records with higher difference	Records with lower difference (%)
0	5530	0
0.05	1202	78.26401447
0.1	648	88.28209765
0.2	350	93.67088608
0.3	166	96.99819168
0.4	65	98.82459313
0.5	18	99.67450271
0.6	1	99.98191682
0.7	0	100

Table 5.23: Noise probability testing result for Broad Oak Farm testing data

Difference(dBA)	Records with higher difference	Records with lower difference (%)
0	7180	0
0.05	1806	74.84679666
0.1	860	88.02228412
0.2	403	94.38718663
0.3	200	97.21448468
0.4	114	98.41225627
0.5	48	99.33147632
0.6	18	99.74930362
0.7	3	99.95821727
0.8	1	99.98607242

Table 5.24: Noise probability testing result for Kell House Farm testing data

total. It is obvious that nearly half of considered points can not satisfy the validation requirement of at least 3 records in their adjacent intervals. For Broad Oak Farm station, we got 11060 rows in total.

For probability calculation, larger data set is preferred. Here, we adopt Broad Oak Farm data to establish the neural network for probability prediction because of its size. To test a trained network against data from the same site, we separate the data set into two sets, one set for training and another set for testing. Each set has 5530 samples. The networks have 4 inputs: distance, operation mode, referencing noise level and thrust. The output is the probability for the noise level of a location is higher than the referencing noise level. We still use 10 sets of inputs as the compound inputs. With many trivial tests, a network is established with two hidden layers consisted of 8 hidden nodes in the first hidden layer and 3 hidden nodes in the second hidden layer. The test results are shown in Table 5.23 and 5.24. Figure 5.27 and 5.28 give the relationship between network output errors (probability difference) and the probability of a testing sample.

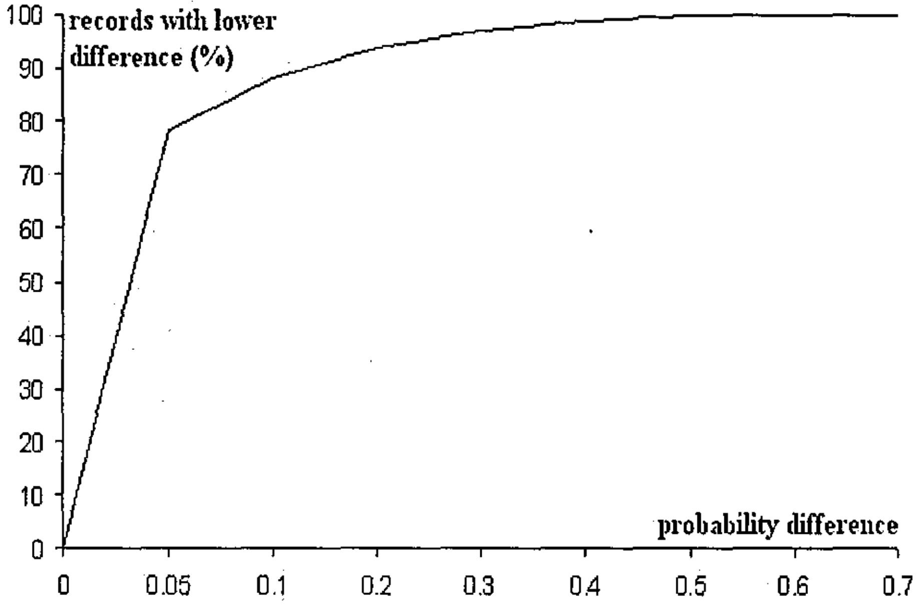


Figure 5.27: Noise probability prediction errors for Broad Oak Farm testing data

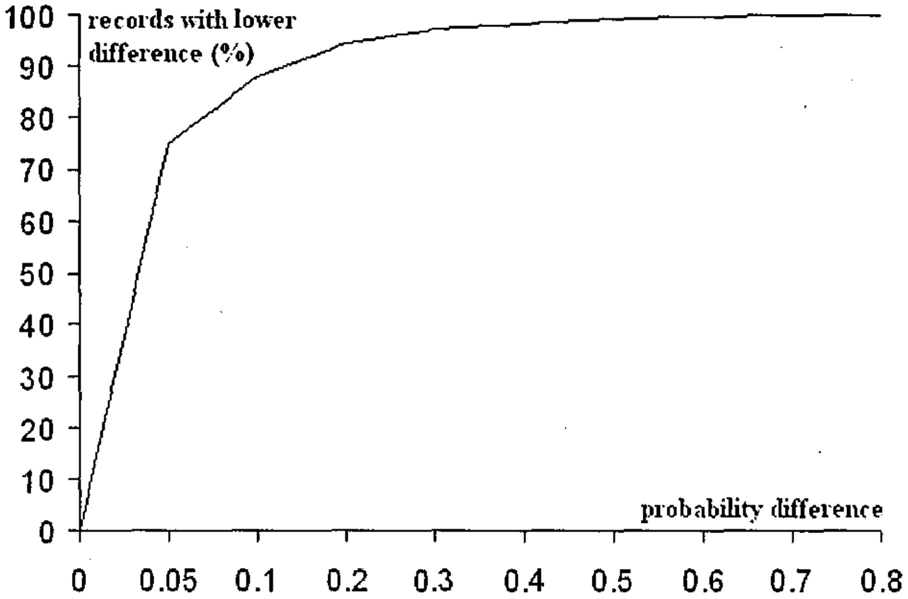


Figure 5.28: Noise probability prediction errors for Kell House Farm testing data

Comparing Table 5.23, Figure 5.27 and Table 5.24, Figure 5.28, it is clear that the noise probability prediction of the trained neural network has similar accuracy in both sites although the data measured in these two sites are independent to each other. Figure 5.29 and 5.30 give the comparison of the independent probability distribution for approaching flights between the two sites with respect to distance under the reference thrust of 4229.42 pounds.

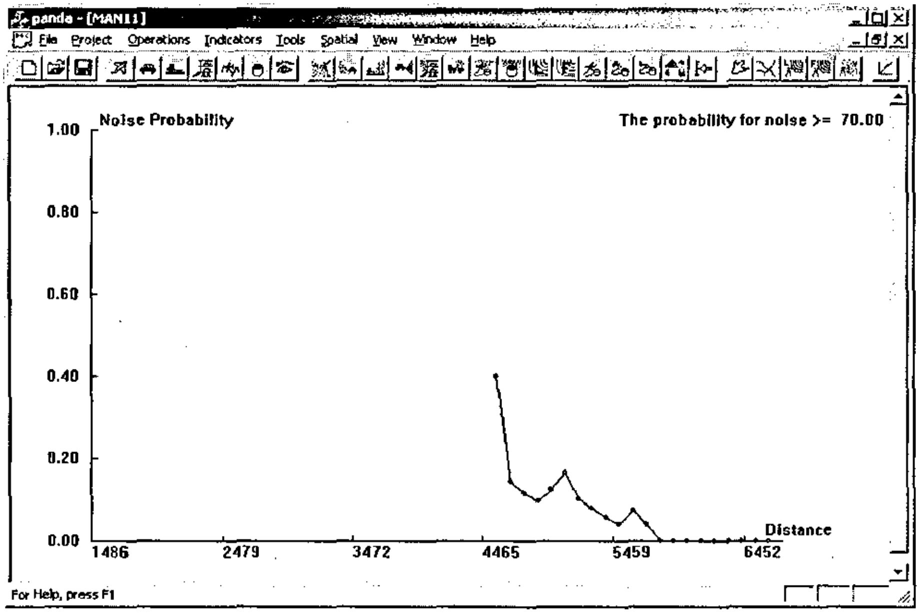


Figure 5.29: Probability for approaching flights at Kell House Farm with respect to distance under the reference thrust of 4229.42 pounds

Based on our experiments on the noise prediction at airports using neural networks trained with in-situ data, it is clear that neural networks provide a convenient tool in adjusting the standard NDP curves to local conditions and hence a network trained with in-situ data can give better prediction than models based on standard NDP curves. Limited by our data availability, we derive thrusts through a reverse mapping neural network simulating standard NDP curves. Such a methodology is bound to enforce a standard NDP curve to the relationships between thrusts and noise levels, and it can increase the accuracy of NDP prediction in reality. Similar phenomena exist including between the derived thrusts and in-situ noise data. However, the relative difference between the standard NDP prediction and neural network prediction is at least the same as shown in this research. The superiority of neural networks over standard NDP model at a local airport is obvious.

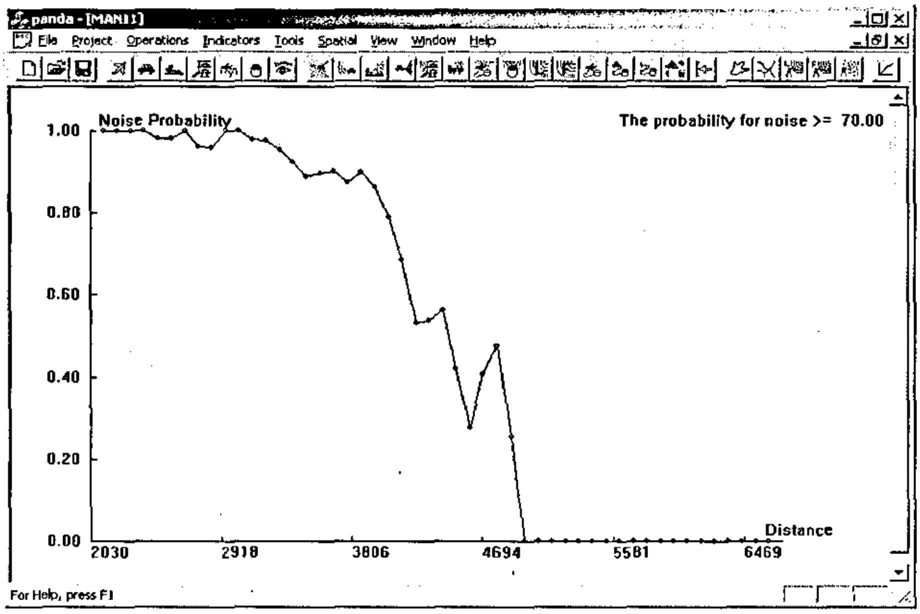


Figure 5.30: Probability for approaching flights at Broad Oak Farm with respect to distance under the reference thrust of 4229.42 pounds

Chapter 6

Prototype

6.1 Structure of the prototype

There are two different strategies to model the airport environment for given operational scenarios: firstly, to model each indicator separately using individual models (the usual approach adopted), or secondly, to integrate them to establish a specific environmental evaluation model. The first strategy is relatively simple, but involves large amounts of work in data input and interpretation. The same data would have to be repeatedly input to different systems in different formats. The bias and errors in the general model from the different geographical and social environments are difficult to consider in this way. Furthermore, this work can easily result in new errors and uncertainties. One of the important tasks for any decision support system is to relieve the user from this kind of work and increase the reliability of the data input and output. Therefore, we highlight the second approach for our airport environment decision support system here – a specific model for the sustainable development of an airport.

Because of the dynamic features of the environmental indicators and the relatively stable mechanisms for pollution calculation and their parameters (taken from the manufacturers of power plant and vehicles), the basic system is separated into two parts: an open database and an inference engine. The open database is for the standard parameters like emission coefficients for different power plant and the user defined operation profile data. As a facility for data mining, the inference engine serves for deriving the general and airport specific environment indicators. For most of the currently available models, the user could input only some profile data and cannot change other data in the database. However, the understanding of the indicator itself is still under development and many relevant parameters are not known at this moment, hence an open structure is a better solution for possible future expansion or updates. Therefore, the proposed system provides complete user control over the whole database part. The user controlled database drives the inference engine to retrieve, calculate and evaluate different indicators according to airport operational scenarios, environmental science and data mining from databases. According to the results of the inference engine, the user could modify the operational data to examine different consequences for different development strategies. In this way, the database and inference engine interact with each other in accordance to the user's will - as shown in Figure 6.1.

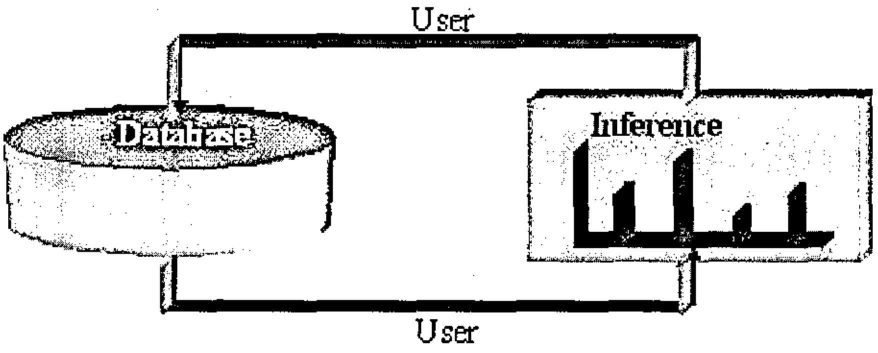


Figure 6.1: Two main components of the system: database and reference engine

Both the database and indicator engine are composed of many different sub models. With respect to their different functions, the framework of the whole system can be illustrated further in Figure 6.2.

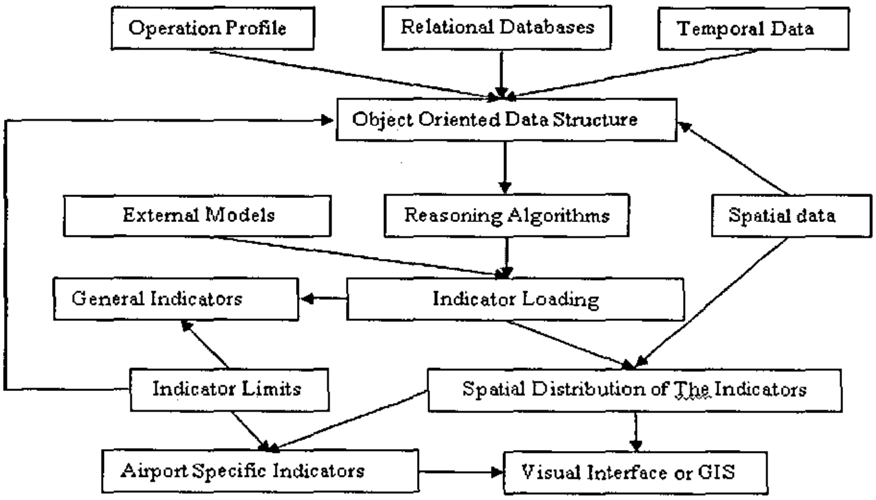


Figure 6.2: The system structure

Here the highlighted boxes represent those components constituting the database, or the open part of the system where the user has full control of its contents. Other boxes form the inference engine.

The open database gets information from the user and drives the operation of the inference engine. It is divided into six parts according to the different functions: operational profile, relational databases, temporal data, spatial data, indicator limits and object oriented database. The operational profile incorporates temporal operational detail. By defining different operational scenarios, the user is able to check their consequences from the point of view of their impact on the airport environment. The relational databases in Figure 6.2 represent those aircraft and vehicle data provided by their manufacturers and the indicators of current interest, such as the coefficients for engine emissions, aircraft noise curves, selected indicators, etc. Because of the dominance of relational databases in industry, most monitoring data also belong to this category. The temporal data consists of the infrastructure modifications proposed at some specific time in the future, such as a new runway, new terminals and so on. The spatial data part serves for the spatial definition of the infrastructure. The different temporal and spatial arrangements could be defined and modified through the two relevant interfaces. The indicator limits are designed for the user-defined indicator limits. These limits are dynamic and represent different interpretations at different times and places. The last item in the database is an object oriented data structure. This is the internal representation of the airport environment data type.

The inference engine receives information from the database, makes various data mining operations in reasoning and calculations and then sends out output indicator values. It consists of 7 different parts with respect to their different functions: reasoning algorithms, external models, loading of various indicators, spatial distribution of the indicators, visual interface or GIS, general and airport specific indicators. The reasoning algorithms play a significant role in the reference engine system. It serves to calculate the environmental indicator loaded from the database using different algorithms. The external models refer to those existing models, such as INM, EDMS, etc. The user has the option to call external models as an alternative to the internal calculations. After the operation of the reasoning engine, the loading of various indicators are obtained and recorded in the indicator loading part. If necessary, a more detailed and airport specific analysis is available through the "spatial distribution of indicators" part and GIS part. By means of the "spatial data" designation, the user is free to design different areas or points as interested objects. The land use and residential influences could be considered using GIS.

Following this idea, a prototype was developed for an airport environment decision support system. The system consists of three sub systems: full functional neural network tool, a full functional CAD tool and a prototype of airport decision support system. All software were developed under Microsoft Visual C++ 6.0.

6.2 Implementation of neural networks - NRSE

NRSE means a neural network tool with RSE calculation capability. NRSE provides full functionality for a neural network tool with BP models, this includes data input, data process and mapping, output presentation etc. NRSE could deal with data in different formats: data input from key board, text files, databases and even GIS raster maps. The data processing function can convert different data into the internal objects and then employ neural network algorithms to get the output. Depending on the requirement, the output could be in the form of on screen printing, text files or GIS maps. In addition to the outputs of neural networks, RSE and its relevant values (Section 4.2) of each input node could be calculated from a trained neural network.

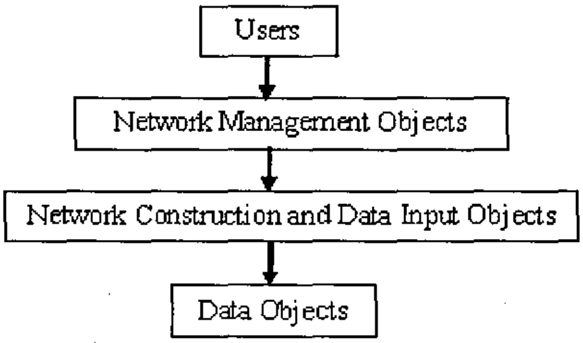

Users
Network Management Objects
Network Construction and Data Input Objects
Data Objects

Figure 6.3: The NRSE structure

Considering efficiency in input and output management, the objects in NRSE are not strictly encapsulated as independent objects according to their different data. The objects are organised with respect to their roles in the input and output of the networks operation. There are three different categories of object classes in NRSE code: objects for data structures, objects for network management and objects for network construction and data I/O (input/output). Their relationship is shown in Figure 6.3. The network management objects are the top layer objects communicating with user. It passes user's requirement or input to network construction and data I/O objects, which constructs neural networks or converts input data into data objects.

Object	Role
CNetNodes	Represent the neural network node values
CNetParameters	Contain all parameters for a neural network
CNetWeights	Record the connection weights of neural networks
CSample	Store the training samples of neural networks
CInputOutputs	Keep original concepts of input and output nodes

Table 6.1: NRSE objects for data structure

6.2.1 Objects for data structure

No matter what data format of the input data, it has to be converted to an understandable data format inside the neural networks to be processed. In NRSE, the data structure are mainly represented as five data objects: CNetNodes, CNetParameters, CNetWeights, CSample and CInputOutputs as shown in Table 6.1.

These data objects are the internal representation of the neural networks in NRSE, and any input data have to be converted into these internal data representation. The sample set for training are converted into CSample first, and then the inputs and outputs are represented in the format of CInputOutputs. Only data in CInputOutputs could be applied into the network and got node values in the form of CNetNodes together with CNetWeights. All these data are encapsulated together by CNetParameters. The roles of these objects in the operation of a neural network is shown in Figure 6.4. Their head file contents are listed in Appendix A.1.

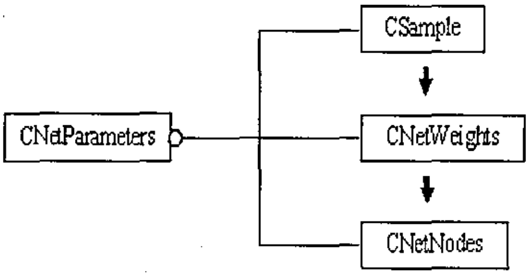


Figure 6.4: The roles of data objects in NRSE

6.2.2 Objects for network construction and data I/O

As a fully functional neural network tool, NRSE provides comprehensive data input and a user friendly network construction facility. To this end, the program implements a series of network construction and data input objects to fulfil this

Object	Role
CAddDatabaseDlg	Add new data from databases
CBmpSampleDlg	Input data from GIS maps
CBpParaDlg	Set the BP environment parameters for a neural network
CGisQueryDlg	Present network mapping result in the form of GIS map
CModifyWeightDlg	Modify the connection weights of networks
COutTableDlg	Present output in the form of a database table
CProcessDlg	Set dynamic parameters before starting training
CSetNetDb	Establish a new neural network from a database
CSetupDlg	Setup a neural network manually

Table 6.2: NRSE objects for network construction and data input

requirement. Compared with other neural network tools, NRSE provides some special input and output facilities. Some of the objects in this category are listed in Table 6.2.

The objects in Table 6.2 can be separated into three different classes according to their roles: network setup, network modification and network I/O, as shown in Figure 6.5. The objects in network setup group are responsible to setup a network structure including the structure of the network and its initial parameters. Having a new network established, its parameters are modifiable through objects in network modification group. A trained neural network will then accept new inputs and give outputs in the required form of users through object in network I/O group. It should be noted that Table 6.2 and Figure 6.5 demonstrate only part of the objects for network construction and data I/O. There are many other objects such as those Page objects for user wizard are not included here for simplicity.

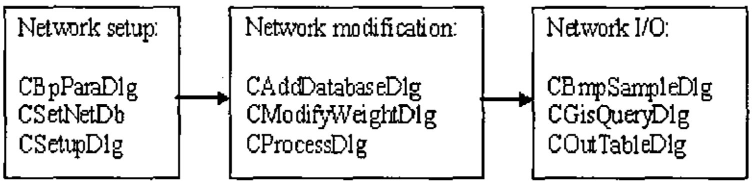


Figure 6.5: Object for network construction and data I/O

6.2.3 Objects for network management

Considering efficiency of the operation, most complicated action functions are directly encapsulated into the network management objects. Compared with the other two groups, management objects has more complicated structure and functions. The two main objects in this group are CNRSEDoc and CNRSEView.

Function Name	Role of Function
ReadGISMapDoc(char* lpszPathName)	Convert GIS map into internal data representation
ReasonBP()	Map input data to output data without RSE values
ReasonRse(int)	Map input data to output data with RSE values
DoBpLearn(UINT m_CurSamNo, BOOL IsLearn)	Train a neural network with given samples

Table 6.3: Main functions in CNRSEDoc

Function Name	Role of Function
DrawGisMap(CDC* dc)	Present a GIS map as reasoning results
DrawRse(CDC* dc, int height)	Display RSE and GRSE values
DrawBpReasonResult(CDC * dc, int height)	Show BP mapping results
DrawStructure(CDC* pDC, int height)	Demonstrate the network structure
DrawError(CDC* dc, int height)	Give errors in the current training process

Table 6.4: Main functions in CNRSEView

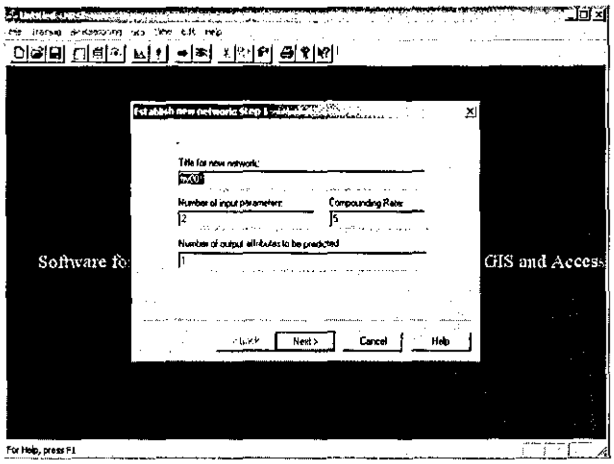
The contents of their head files are shown in Appendix A.2. Among these two objects, CNRSEDoc manages the storage, training and various calculations; whereas CNRSEView is mainly responsible for the I/O and user interaction and communication.

CNRSEDoc is the object holding network data and responsible for network training, mapping and RSE evaluation. The main data member in CNRSEDoc is m_Para, which is a CNetParameters data type holding all network parameters including network nodes, weights and samples as explained in Section 6.2.1. In addition to message map functions and storage function, some significant calculations functions for neural networks are shown in Table 6.3. It includes functions for training process, reasoning with or without RSE evaluation and GIS map operation.

Compared with NRSEDoc, the main responsibility of NRSEView is the presentation of the mapping results in the form of Graphics, GIS maps, Databases tables or text files. In addition to the usual message functions, some typical functions are listed in Table 6.4. CNRSEView holds more data members than CNRSEDoc, but they are mainly parameters for the presentation environment. The details of these parameters could be found in Appendix A.2.2

6.2.4 User interface and functionality

Based on the objects implemented, NRSE provides a user-friendly user interface to its users. A user would not need deep knowledge to setup a neural network and start its training. There is a step by step network setup wizard to get a user through the whole process of establishing a new neural network. Figure 6.6 shows the starting page of the wizard. Different from other neural network tools, the compound model in Section 4.4 has been incorporated into NRSE (as shown in Figure 6.6).



Title for new network: [illegible]

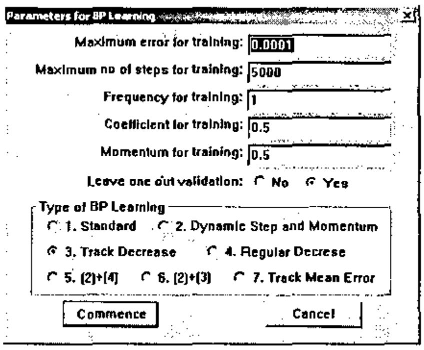
Number of input parameters: 2

Compounding Rate: 5

Number of output attributes to be predicted: 1

Figure 6.6: A wizard for establishing a new neural network

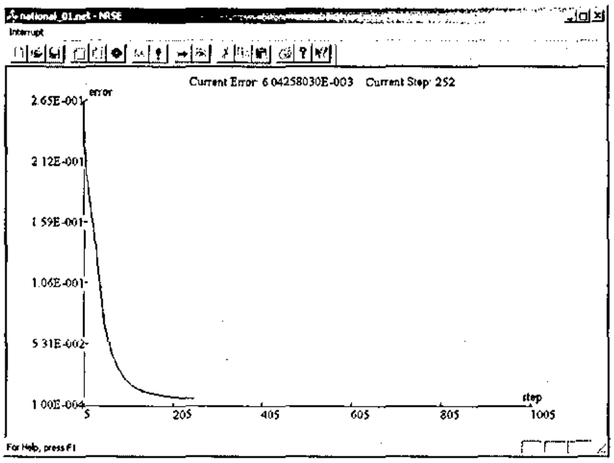
The network can also be established directly from your training data set, and the system can automatically assign those boundary values for each input and output nodes according to your provided training data. It is simple to check the training process and update the training parameters because of the objects in Section 6.2.2. Figure 6.7 demonstrates the dialog box for setting network training parameters before starting the training process. Figure 6.7 shows that NRSE provides many different methods of BP training, including the standard BP training, dynamic step and momentum method, track decrease and regular decrease, track mean errors as well as some combinations. Here, the dynamic step and momentum method, track decrease and regular decrease and track mean errors refer to the weight updating frequency and momentum modification method with respect to the dynamic training error change. The dynamic step and momentum method may automatically increase or decrease it according to the dynamic errors; the track decrease and regular decrease will only decrease it with the reduction of errors; whereas the track mean errors will only keep pace with the mean errors. According to the availability of testing samples, the trained neural network could be validated by independent sample set or "leave one out validation". If the sample set is very limited, then a "leave one out validation" could be selected. Under this situation, a text log file will be created to record the test results for each iteration. In NRSE, a training process can be monitored through a dynamic updated graphic presentation as shown in Figure 6.8. The whole process could be controlled by an advanced error threshold and maximum number of training iterations. In addition to this, the process could also be interrupted anytime by an user through function key. An interrupted training can be easily resumed as well.



Maximum error for training: 0.0001
Maximum no of steps for training: 5000
Frequency for training: 1
Coefficient for training: 0.5
Momentum for training: 0.5
Leave one out validation: ○ No ● Yes

Type of BP Learning
○ 1. Standard ○ 2. Dynamic Step and Momentum
● 3. Track Decrease ○ 4. Regular Decrese
○ 5. (2)+(4) ○ 6. (2)+(3) ○ 7. Track Mean Error

Figure 6.7: A dialog box for setting the training parameters



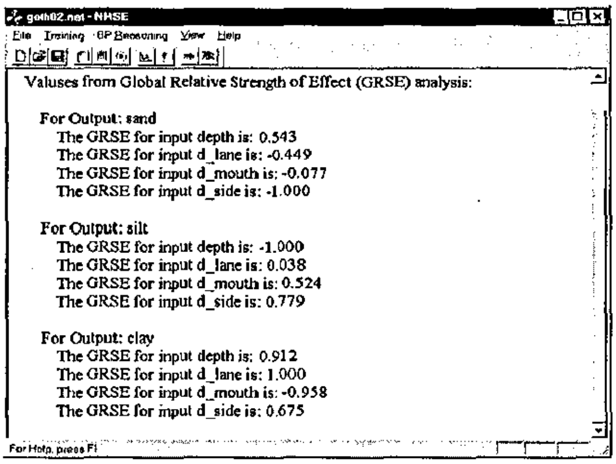
Current Error: 6.04258030E-003 Current Step: 252

Figure 6.8: A dynamic monitoring of the training process

A distinct feature of NRSE compared with other neural network tools is its facility for RSE (GRSE, PRSE and GPRSE) and GIS maps. RSE and GRSE values could be easily calculated from a trained neural network. As a global parameter, GRSE could be derived from the network parameters at any time, but RSE values have to be derived together with given input node values. Because of this, the reasoning operation of NRSE provides two options:

reasoning with or without RSE values. Figure 6.9 demonstrates an output of RSE values. In addition to RSE analysis, NRSE incorporates GIS map analysis function as well, and it could accept Idrisi GIS maps as inputs or training data, and provides facility to give output GIS maps with given input GIS maps. Figure 6.10 is a GIS map created by NRSE from input maps. The map here is presented in Idrisi environment.

NRSE provides a convenient facility for the user to modify network parameters. This includes not only parameters like training momentum and error thresholds, but also the network structures and connection weights. For example, users are free to change the connection weights and fix it while allow other weights to be updated in training. Figure 6.11 shows the dialog box which appears when a corresponding connection weight is double clicked in weights observation window.



```
Valuses from Global Relative Strength of Effect (GRSE) analysis:

  For Output: sand
    The GRSE for input depth is: 0.543
    The GRSE for input d_lane is: -0.449
    The GRSE for input d_mouth is: -0.077
    The GRSE for input d_side is: -1.000

  For Output: silt
    The GRSE for input depth is: -1.000
    The GRSE for input d_lane is: 0.038
    The GRSE for input d_mouth is: 0.524
    The GRSE for input d_side is: 0.779

  For Output: clay
    The GRSE for input depth is: 0.912
    The GRSE for input d_lane is: 1.000
    The GRSE for input d_mouth is: -0.958
    The GRSE for input d_side is: 0.675
```

Figure 6.9: RSE values from a trained neural network

6.3 Implementation of CAD facility

As explained in Chapter 1, a decision support system for airport environment has to consider the third dimension in addition to the usual 2D operation for flight trajectory. Sustainability is usually evaluated against a long term like 20 – 50 years, hence its relevant airport planning involves not only changes in operation details but also some construction changes as well. Therefore, the CAD facility for airport environment evaluation has to provide user-friendly function to give users freedom to change both operation details and construction details in 3D space. To fulfil this requirement, a CAD system was developed for the airport decision support system. Considering its close relationship with

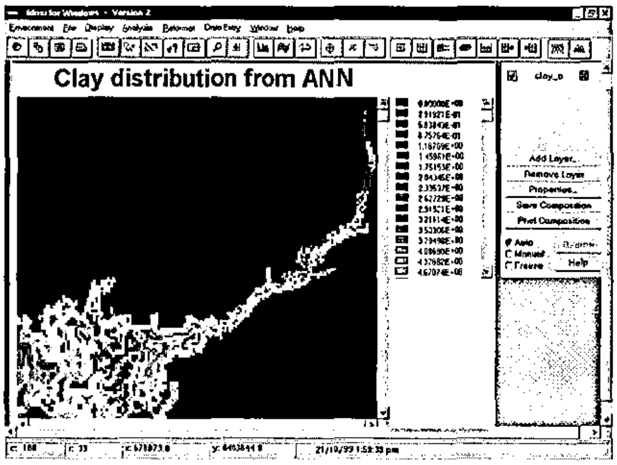
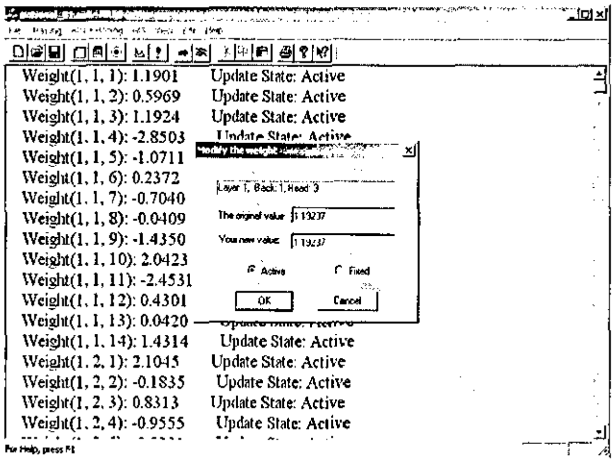


Figure 6.10: A GIS map obtained from NRSE



```
Weight(1, 1, 1): 1.1901     Update State: Active
Weight(1, 1, 2): 0.5969     Update State: Active
Weight(1, 1, 3): 1.1924     Update State: Active
Weight(1, 1, 4): -2.8503    Update State: Active
Weight(1, 1, 5): -1.0711
Weight(1, 1, 6): 0.2372
Weight(1, 1, 7): -0.7040
Weight(1, 1, 8): -0.0409
Weight(1, 1, 9): -1.4350
Weight(1, 1, 10): 2.0423
Weight(1, 1, 11): -2.4531
Weight(1, 1, 12): 0.4301
Weight(1, 1, 13): 0.0420
Weight(1, 1, 14): 1.4314    Update State: Active
Weight(1, 2, 1): 2.1045     Update State: Active
Weight(1, 2, 2): -0.1835    Update State: Active
Weight(1, 2, 3): 0.8313     Update State: Active
Weight(1, 2, 4): -0.9555    Update State: Active
```

Modify the weight

Layer 1, Back: 1, Head: 3

The original value: 1.13237

Your new value: 1.13237

● Active ○ Fixed

Figure 6.11: Facility for network connection weight modification

Function Name	Role of Function
GetDistance(CRealPoint& p, int dimension)	Calculate distance between two points
GetShortDisToSegment(CRealPoint p1, CRealPoint p2, int dimension)	Find the shortest distance between a point and a segment of line
Rotate(double angle, double xc, double yc)	Rotate the coordinates of a point with a given angle
GetDisPtoLine(CRealPoint p1, CRealPoint p2, int dimension)	Get the shortest distance between a point and a line
ScreenToReal(CMapScale& scale)	Convert the coordinates from screen system to real world system

Table 6.5: Main functions in CRealPoint

the airport environment prototype, this system was embedded into the airport environment system rather than a standalone system like NRSE.

There are mainly three core objects in the CAD facility: CRealPoint, CJoint and CEntity. CRealPoint encapsulates the data defining a 3D point and its relevant functions. CJoint represents a joint point between lines. CEntity encapsulates all data and functions related with geometrical entities. Their full lists are shown in Appendix B.

CRealPoint is the basic data structure for a point in the CAD facility. In addition to the three coordinates m_X, m_Y and m_Z, it provides two additional parameters: m_M for uncertainty and m_Sign for its attribute control mark. Here, m_Sign is a conserved parameter for any control use in program. However, m_M is a significant feature of CRealPoint. It means that CRealPoint will be able to keep not only the three coordinates as a result of measurement, but also its error or uncertainty. Here, we use the grey geometry point as its representation: the three coordinates m_X, m_Y and m_Z represent its core point, and the m_M denotes its grey radius.

CRealPoint provides the cell component for complex geometrical shapes. All shapes, including points, line segments, squares, polygons, circles, ellipses and complex rings, can be represented with a set of organised points. Here, CEntity is the object to encapsulate all these shapes and their corresponding operations. Compared with CRealPoint, CEntity has much more complicated data members and functions. There are 21 data members in CEntity, and the most important members are m_PointsSet, m_Sort, m_SubEntity and m_Uncertainty, as shown in Table 6.6.

For a simple entity, such a polygon, m_PointsSet holds all the points in the form of an array of CRealPoint. For complex entity, such as a ring, m_SubEntity

Name	Data Type	Role
m_PointsSet	CArray<CRealPoint, CRealPoint>	A set of points constituting the shape
m_Sort	int	A mark for different shapes
m_SubEntity	CTypedPtrList<CPtrList,CEntity*>	A set of entities constituting a complex entity like ring
m_Uncertainty	CArray<double,double>	A set of parameters for uncertainty

Table 6.6: Main data members in CEntity

holds a set of entities which hold their own points. The key in identifying different entities is m_Sort. Depending on the value of m_Sort, different geometrical shapes could be constructed using CEntity, such as points, line segments, Bezier curves, polylines, polygons, rectangles, ellipses, arcs, rings and compound shapes. Some typical shapes created using CEntity is shown in Figure 6.12.

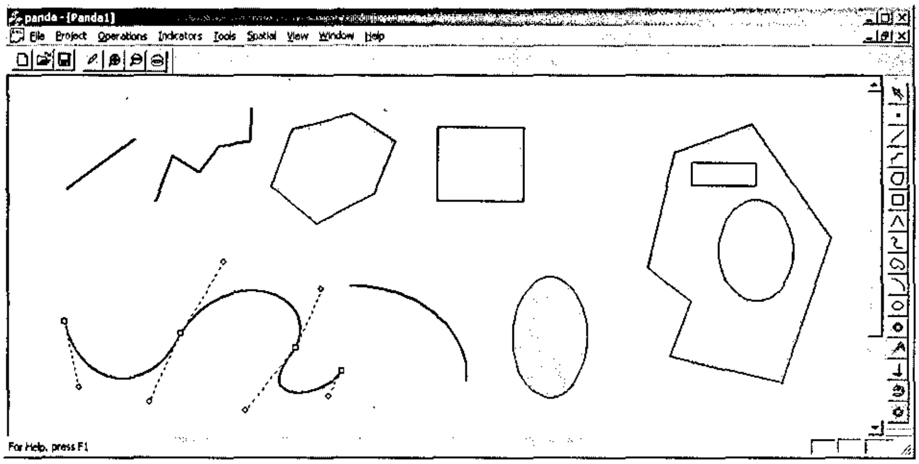

panda - [Panda1]
File Project Operations Indicators Tools Spatial View Window Help
For Help, press F1

Figure 6.12: Geometrical shapes created using CEntity

CEntity encapsulates all operations for an entity into the object. Considering the variety and complexity of operations for an entity, there are more functions in CEntity and they are much more complicated than CRealPoint. Depending on m_Sort, these operation may operate on m_PointsSet or recursively operate CEntity through m_SubEntity, as shown in Figure 6.13. Some functions are demonstrated in Table 6.7.

There are two functions clearly distinct from other CAD environment: OverlayOperation and GetUncertainty. OverlayOperation is designed for overlay operation in GIS environment. It carries out operations between two objects,

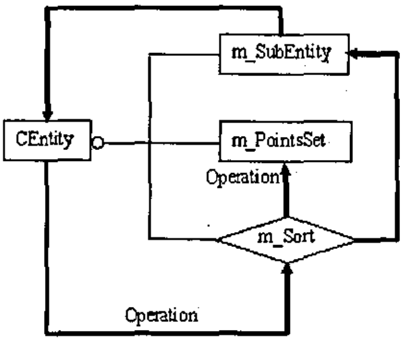


Figure 6.13: Relationship between CEntity operation and m_PointsSet

Function Name	Role of Function
DrawEntity(CDC*, BOOL)	Draw the entity
GetLength(CMapScale&, int sub=0)	Calculate the length of the entity boundary
GetArea(CMapScale&, int sub=-1)	Get the covering area of the entity
FindCenter()	Find the centre point of an entity
GetIntersection(CEntity&, CArray<CJoint,CJoint>&, int)	Derive the intersection points between two entities
PtInEntity(CRealPoint)	Ceck if a point is inside an entity or not
OverlayOperation(CEntity&, CTypedPtrList<CPtrList, CEntity*>&, int, double)	Overlay two entities for AND, OR and DIFFERNCE operation
GetUncertainty(CMapScale& scale)	Compute uncertainty of an entity

Table 6.7: Main functions in CEntity

such as UNION, INTERSECTION, DIFFERENCE and simple overlay. Figure 4.18 and Figure 4.19 present two operand objects, and Figure 4.20 demonstrates their DIFFERENCE. GetUncertainty is responsible for uncertainty evaluation. As aforementioned, each CRealPoint object stores its uncertainty in its m_M member, and this information provides the basis for GetUncertainty to derive the uncertainty of each shape object. Therefore, the uncertainty of each shape could be checked using GetUncertainty function. This functionality provides a powerful capability for the airport environment information system to evaluate its uncertainty associated with map data. It is significant especially for those GIS related system where mass overlay operations are routine work. Those overlay operation can easily bring together objects with different uncertainty, this make the new object show different uncertainty from any of its operands. With CEntity, all information on uncertainty is kept in CRealPoint, and it will be transfered to the new object no matter how many overlay operations have been carried out. Therefore, its uncertainty information will be kept as well.

Table 6.7 presents only part of the available functions of CEntity, and the full list is provided in Appendix B.2. CEntity provides convenient tools to input, modify and edit maps. All processes could be simply realised on the screen by moving and clicking mouse. For example, a complicated geometrical shape could be assembled by creating its parts first, and then assembled together by a "stick" operation. It makes the design and modification of airport structure an easy task, especially considering the frequent change during a long term operation scenario.

6.4 Implementation of airport model – PANDA

For an airport environment evaluation model, the most important indicators to be evaluated are those indicators about emission, noise, water consumption and waste. All these relate to the operation of airports. Therefore, the model has to consider both operation side and environment side together. For a long term evaluation, analytical model is useful in most cases, and a detailed spatial model would be useful when some snapshot is necessary for some specific time points. In this sense, both analytic model and spatial model are useful. To this end, PANDA is implemented as a model to facilitate both analytic and spatial analysis.

6.4.1 Objects of PANDA

As a prototype of an airport environment evaluation model, objects are used to model the components of airports. Depending on the complexity of objects,

Object	Role
CAircraftEmissionLinkData	Link aircraft emission databases with aircraft in the airport
CAircraftGenOptData	Contain data for general airport operation details
CAircraftOperationData	Keep data for aircraft operation
CFugitiveData	Store data for airport fuel tanks
CIndicatorLimits	Hold user defined limits of environment indicators
CInfrustructure	Record data for airport infrastructure
CNoiseData	Relate aircraft operation with its manufacturer provided noise data
CPassengerNumber	Give the passenger numbers of each transportation mode and their ratio
CRunwayData	Hold runway data for the airport
CServiceGseData	Hold service related fugitive emission sources data
CSurfaceData	Keep data for surface transport
CWasteWaterElectricityData	Store all data about waste, water and electricity at an airport
CWeatherData	Keep weather data at an airport
CHubAirport	Hold all data for airport operation and environment

Table 6.8: PANDA objects for data structure

some objects provide only data structure, and some other objects encapsulate both data and functions together. Similar to NRSE, the objects in PANDA can be classified into three different groups: data objects, model construction and I/O objects, management objects.

Data objects

PANDA has to represent those airport components contributing to the environment problem. It includes all the airport infrastructure components and operation details need to be modeled in PANDA. Here, a specific data structure is constructed to represent different components and operation details. Some of the data objects are listed in Table 6.8. It should be noted that PANDA includes all those objects in Table 6.1, CEntity and CRealPoint in addition to the objects in Table 6.8. The reason is that PANDA integrates these together.

Due to the complexity of an airport infrastructure and operation, each data object contains many data members. Some objects are basic data objects which provide only a data structure for relevant data representation, some other objects are advanced objects involving other object as their members and complicated functions. In Table 6.8, CAircraftEmissionLinkData, CAircraftOperationData, CFugitiveData, CIndicatorLimits, CInfrustructure, CNoiseData, CPassengerNumber, CRunwayData, CServiceGseData, CSurfaceData, CWasteWaterElectricityData and CWeatherData are basic objects; whereas CHubAirport and CAircraftGenOptData are advanced objects. Both CAircraftOperationData and CPassengerNumber are members of CAircraftGenOptData, and all objects are direct or indirect members of CHubAirport. Their relationship is shown in Figure 6.14

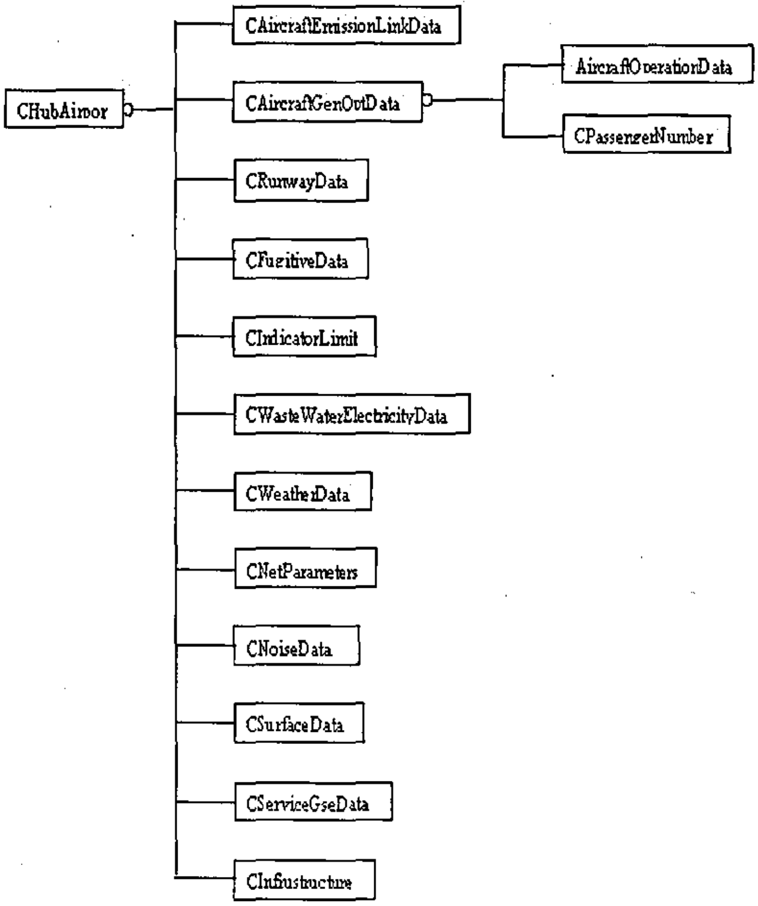


Figure 6.14: Relationship between data objects of PANDA

Compared with basic data objects, the advanced objects are more complicated with their associated data members and functions. CHubAirport is the main data object encapsulating most data and functionality in PANDA. Some of its associated data members are listed in Table 6.9, and some of its representative functions are listed in Table 6.10. For the full list of data members and functions of each object, please refer to Appendix C.1.

Similar to other airport environment information systems, Panda is associ-

Name	Data Type	Contents
m_SusIndicatorValueSet	double**	A set of values for sustainable development indicators;
m_SusIndicatorNameSet	CStringArray	A set of names for sustainable development indicators;
m_WasteData	CWasteWaterElectricityData	Data about waste;
m_WaterDatay	CWasteWaterElectricityData	Data about water consumption;
m_ElectricityData	CWasteWaterElectricityData	Data about electricity consumption;
m_Weather	CWeatherData	Data about weather condition;
m_AnnLevelAircraft	CStringArray	Data about aircraft noise level from neural networks;
m_AnnProbaAircraft	CStringArray	Data about aircraft noise probability from neural networks;
m_Ann	CNetParameters	Data about established neural networks;
m_NoiseLimit	float	The limit noise value;
m_NoiseCurve	CArray<double,double>	Data about noise boundary;
m_NoiseData	CNoiseData	Data about noise;
m_MotorwayData	CArray<CSurfaceData,CSurfaceData>	Data about motorway around an airport;
m_CarParkLinkData	CArray<CSurfaceData,CSurfaceData>	Data about roads linking car parks near an airport;
m_CarParkData	CArray<CSurfaceData,CSurfaceData>	Data about car parks near an airport;
m_RailData	CArray<CSurfaceData,CSurfaceData>	Data about rails near an airport;
m_ServiceGseData	CServiceGseData	Data about airport fuel service;
m_IndicatorLimits	CIndicatorLimits	Data about user defined indicator limits;
m_FugitiveData	CArray<CFugitiveData,CFugitiveData>	Data about fugitive emission at an airport;
m_RunwayData	CTypedPtrList<CPtrList,CRunwayData*>	Data about runways;
m_RunwayUsageData	CRunwayUsageData	Data about runway usage;
m_AircraftOptData	CAircraftGenOptData	Data about aircraft operation at an airport;
m_EmissionLink	CAircraftEmissionLinkData	Data about the links between aircraft and emission databases;
m_Emission	double***	Data about emission at an airport;
m_Infrustructure	CArray<CInfrustructure,CInfrustructure>	Data about airport infrastructure;
m_NumEmissionYear	int	Number of years for the evaluation of emission accumulation

Table 6.9: Main data members in CHubAirport

ated with a series of standard databases as its references. It includes databases for aircraft parameters such as engine, weight and passenger numbers, car park factors databases for car park emissions, car park operation databases, cold start databases, electricity, water and waste databases, GSE databases for fugitive emissions, ICAO data bank for aircraft engine exhaust emissions etc. All these databases serve as a reference for the relevant data objects to derive their data using the standard databases.

6.4.2 Objects for model construction and I/O

Although PANDA is only a prototype, it is necessary to provide some input and output functionality to make the system convenient for its user to use it. To this end, a series of objects for model construction and I/O operation are implemented. These objects provide model construction interface using wizard, and direct link with MSAccess database for data input. The output of the model could be visualised as text output or graphics. Some of the relevant objects are listed in Table 6.11. It should be noted that not all objects in this group are shown in this table.

With these objects, users have freedom to update any data at any stage. The user defined infrastructure, operational scenarios and other related data will be kept in the database as objects, and those scalar data from external relational database will still be kept in their corresponding relational databases. They will be called in only when it is necessary, such as neural network training and historical statistical analysis. Objects in Table 6.11 will keep the linkage between relational database and the object oriented data structure. This linkage could be modified by its user at any time, as shown in Figure 6.15.

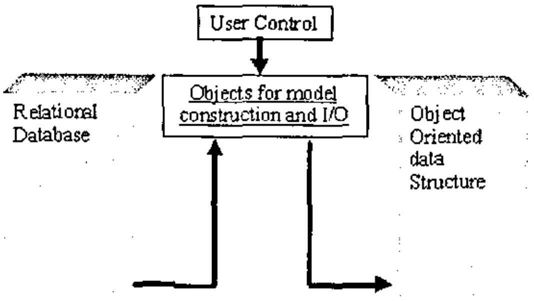

User Control
Objects for model construction and I/O
Relational Database
Object Oriented data Structure

Figure 6.15: Relationship between objects of PANDA and relational databases

Function Name	Role of Function
InsertInfrustructrure(CEntity&, CMapScale&)	Add in new airport infrastructure;
ModifyInfrustructure(CString; CEntity&, CMapScale&)	Modifying existing infrastructure;
CheckCapacity()	Check airport operational capacity;
GetGeneralNoise(BOOL)	Find general noise level and frequency probability;
GetTotalIndicatorValue(double**)	Evaluate the indicators to give a general evaluation;
GetSusIndicators()	Calculate the indicators of sustainable development;
GetWasteWaterElectricityConsumption()	Compute waste, water and electricity consumption;
GetSurTrafficFuelConsumption(BOOL)	Evaluate the fuel consumption of surface vehicles;
GetAirFuelConsumption()	Find fuel consumption of aircraft operation;
GetCarparkEmission()	Estimate car park emission;
GetEmissionDispersion(CRealPoint, CRealPoint, double)	Disperse emission in at an airport
AnnReason(CStringArray&, CSample&)	Call neural network to get indicator values
GetNoiseDataStatistics(CString, CString, CString, double, double)	Calculate statistics of monitored noise data;
GetFugitiveEmission()	Evaluate fugitive emission at an airport;
GetGseEmission()	Derive airport fuel tank emission;
GetAirportServiceEmission()	Estimate airport service related emission;
GetColdStartEmission()	Evaluate cold start emission;
GetSurTransEmissions(BOOL)	Calculate emission from surface transportation near an airport;
DrawIndicators(CDC*, int, int, int, CString, CRect&, int, int, CPoint, CStringArray* monitor=NULL)	Visualise indicator values;
GetAircraftEmission()	Calculate aircraft emission;

Table 6.10: Main functions in CHubAirport

Object	Role
CAircraftEmissionSetupDlg	Setup the link between aircraft emission databases and aircraft engine;
CCompRateDlg	Define the composition of aircraft fleet at an airport;
CFugitiveEmissionDlg	Input data for airport fuel tanks;
CIndicatorLimitDlg	Define limits of environment indicators;
CInfrustructureDlg	Design or input airport infrastructure;
CMonitoringDataDlg	Input or link airport monitoring data;
CNoiseAnalysisDlg	Setup parameters for noise analysis;
CNoiseDataDlg	Input noise related data;
CNoiseDistributeDlg	Define the noise distribution parameters;
CProjectSetting	Establish project environment;
CRunwayOptValueDlg	Define runway operation data;
CRunwayUsageDlg	Design ratio of operation among runways;
CSurVehicleDistributeDlg	Distribute surface vehicles on roads;

Table 6.11: PANDA objects for model construction and I/O

6.4.3 Management objects

Similar to NRSE, there are mainly two management objects in PANDA: CPandaDoc and CPandaView. It is at this level that PANDA integrates all objects in the neural network systems, CAD facility and airport systems. CPandaDoc is responsible for data integration management and storage, and CPandaView manages all the interaction between different objects and users. Both CPandaDoc and CPandaView contain a number of data members, the data members in CPandaDoc serve for the persistent storage of the model, but the data members of CPandaView are mainly active only in the operation stage of the model. Some of the representative data members of the two objects are shown in Table 6.12 and 6.13. It should be noted that most data members of NRSE and CEntity are also included in CPandaDoc and CPandaView, and they are not included in Table 6.12 and 6.13. In Table 6.12, the most important data members are m_Airport and m_EntitySet. m_Airport is an object of CHubAirport and holds all data relevant to an airport environment. m_EntitySet is a set of entities of CEntity, which represent the spatial infrastructure of an airport. PANDA is able to run analytical analysis using m_Airport without consideration of spatial details. In the same time, it can also make spatial analysis by means of a combination of m_Airport and m_EntitySet. In Table 6.13, the most important data members are m_Indicator and m_CurrentEntity. m_Indicator is the current active indicator under consideration, and all operations are working on this specific indicator. m_CurrentEntity indicates the current active entity in consideration, so any spatial operation is related to this entity. Both members

Name	Data Type	Contents
m_dbName	CString	Name of associated database;
m_Airport	CHubAirport	Data for an airport;
m_FieldsType	CStringArray	Database field data types;
m_DatabaseFields	CStringArray	Database field names;
m_DocName	CString	Model name;
m_BkgImage	CBkgImage	Data for background image;
m_Scale	CMapScale	Scale of real data to screen size;
m_EntitySet	CTypedPtrList<CPtrList, CEntity*>	A set of spatial entities constituting the airport;

Table 6.12: Main data members in CPandaDoc

are directing the focus point to specific target in the relevant airport.

Because of the encapsulation of CEntity and CHubAirport, the operation functions of CPandaDoc and CPandaView are mainly for data storage and view management. In addition to this, the usual message mapping functions are parts of the two objects as well. Table 6.14 and 6.15 demonstrate some representative functions in these two objects. Table 6.14 shows that PANDA can manage not only its own data, but also data from ArcView and data from relational databases like MSAccess. This functions provides powerful capability for the user to modify and access these data. Table 6.15 demonstrates the functions for user to get dynamic output for any required airport location or map for a large area. These functions are essential for users to check WHAT IF scenarios.

6.4.4 User interface and functionality

The functionality of PANDA is mainly for analysing the sustainability of a proposed airport development plan, such as a new runway, a new terminal, increase of flights or passenger number etc. For a proposed new development of an airport, PANDA is a prototype to evaluate its potential effect on the airport environment such as noise and emission. This functionality is illustrated in Figure 6.16.

The model is established using user provided airport data, vehicle and aircraft data from standard relational databases, airport monitor data and neural

Name	Data Type	Contents
m_MonitorSet	CStringArray	A set of monitoring points;
m_NonSpatialDrawAction	int	An indicator for output presentation;
m_Indicator	CString	The current active indicator;
m_ZValueEdit	CZValueEdit*	A pointer to Z value edit panel;
m_Selected	CArray<int,int>	An array of numbers representing the selected entities;
m_CurrentEntity	CEntity	The current selected entity;

Table 6.13: Main data members in CPandaView

Function Name	Role of Function
UpdateInfrustructure(CArray<CEntity*,CEntity*>&)	Modify the infrastructure at an airport;
OpenAttachedFiles(LPCTSTR)	Open a model file;
SaveAttachedFiles(LPCTSTR)	Save a model file;
ReadDatabase()	Read data from a database;
SaveDatabase()	Save data into a database;
OpenArcView(LPCTSTR)	Open a file in ArcView format;
ShowEntityXYData(int)	Display the geometrical data of an entity;

Table 6.14: Main functions in CPandaDoc

Function Name	Role of Function
GetGridStructure(int&,int&,int&,int&,int&,int&)	Calculate grid structure;
GetEmissionMonitorValue()	Find emission at a monitoring position;
GetNoiseMonitorValue(BOOL)	Get noise values for a monitoring position;
DefineEntityProperty(CEntity*)	Modify the properties of an entity;
DrawThemeMap(CDC*)	Draw the theme map of required airport area;
GetContour(CDC*, double, CArray<double,double>&, int, COLORREF, BOOL)	Calculate a contour for a given indicator;
SelectEntity(CPoint)	Find selected entity according to given point;

Table 6.15: Main functions in CPandaView

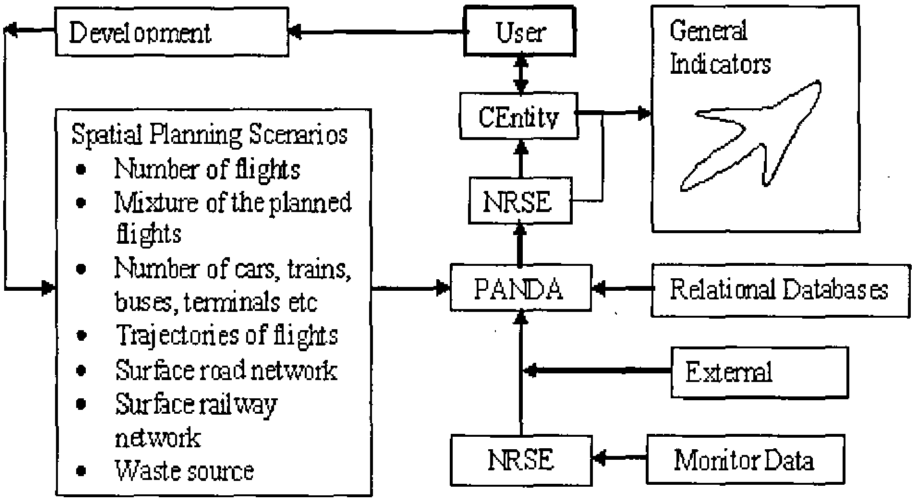
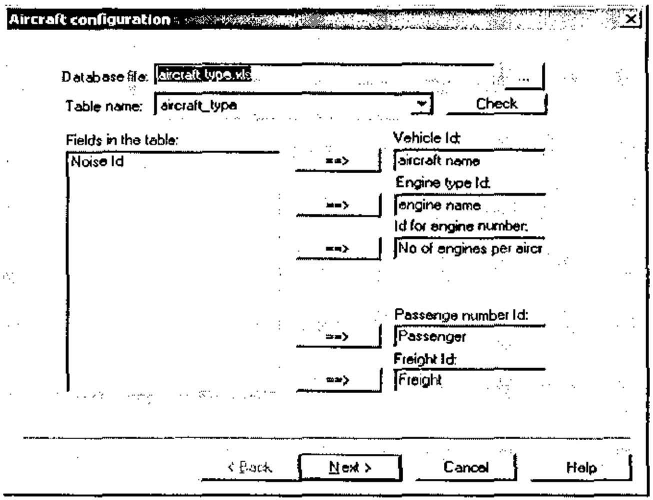

Development
User
General Indicators
CEntity
Spatial Planning Scenarios
• Number of flights
• Mixture of the planned flights
• Number of cars, trains, buses, terminals etc
• Trajectories of flights
• Surface road network
• Surface railway network
• Waste source
NRSE
PANDA
Relational Databases
External
NRSE
Monitor Data

Figure 6.16: Airport development evaluation using PANDA

networks. After a model has been established, then a series of WHAT IF scenarios about airport planning could be tested in the model, such as a new runway next year, a significant increase of flight numbers etc. This test will bring some results on the concerned indicators, and then a comparison could be made between different scenarios. To this end, a user-friendly interface has a crucial role in this analysis. Based on the implemented objects in PANDA, an interface with an "easy to use" feature has been established. A user wizard will guide the user through the different stages of establishing the model. Figure 6.17 demonstrates the first page for a wizard to establish a model.



Aircraft configuration

Database file: aircraft_type.xls

Table name: aircraft_type

Fields in the table: Noise Id

Vehicle Id: aircraft name

Engine type Id: engine name

Id for engine number: No of engines per aircr

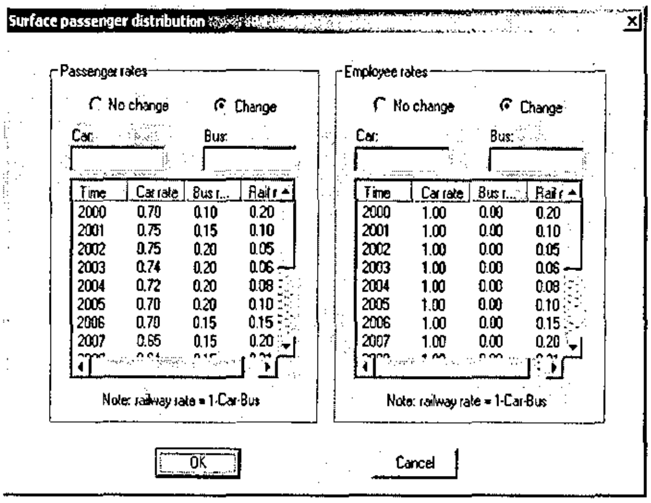
Passenge number Id: Passenger

Freight Id: Freight

Figure 6.17: A wizard for establishing an airport model in PANDA

Having established the airport model, it is easy to modify it any any stage. Figure 6.18 illustrate a dialog box for modifying the distribution of passengers and employees in different transport modes.

In addition to data input through a dialog box, spatial data can also be inserted directly by mouse click on screen or using databases. All functionality of CEntity is available for PANDA. Figure 6.20 shows an a runway, its trajectories and some surface roads constructed through CEntity. In Figure 6.20, the tool bar in the right side provides convenient tools for inputting a new airport related infrastructure, such as runways, terminals, tanks, roads and trajectories. The 3D facility in CEntity provides PANDA a powerful capability in trajectory design. Figure 6.19 shows the selected trajectory's third dimension in a specific pop up window. The third dimension values could be visually modified through this window. The top tool bar in the window present quick access to airport analysis functions. For example, noise analysis will give the distribution of noise around the airport using neural networks, as shown in Figure 6.21.



Time	Car rate	Bus r...	Rail r
2000	0.70	0.10	0.20
2001	0.75	0.15	0.10
2002	0.75	0.20	0.05
2003	0.74	0.20	0.06
2004	0.72	0.20	0.08
2005	0.70	0.20	0.10
2006	0.70	0.15	0.15
2007	0.65	0.15	0.20

Note: railway rate = 1-Car-Bus

Time	Car rate	Bus r...	Rail r
2000	1.00	0.00	0.20
2001	1.00	0.00	0.10
2002	1.00	0.00	0.05
2003	1.00	0.00	0.06
2004	1.00	0.00	0.08
2005	1.00	0.00	0.10
2006	1.00	0.00	0.15
2007	1.00	0.00	0.20

Note: railway rate = 1-Car-Bus

Figure 6.18: A dialog box for modifying passenger and employee distribution in PANDA

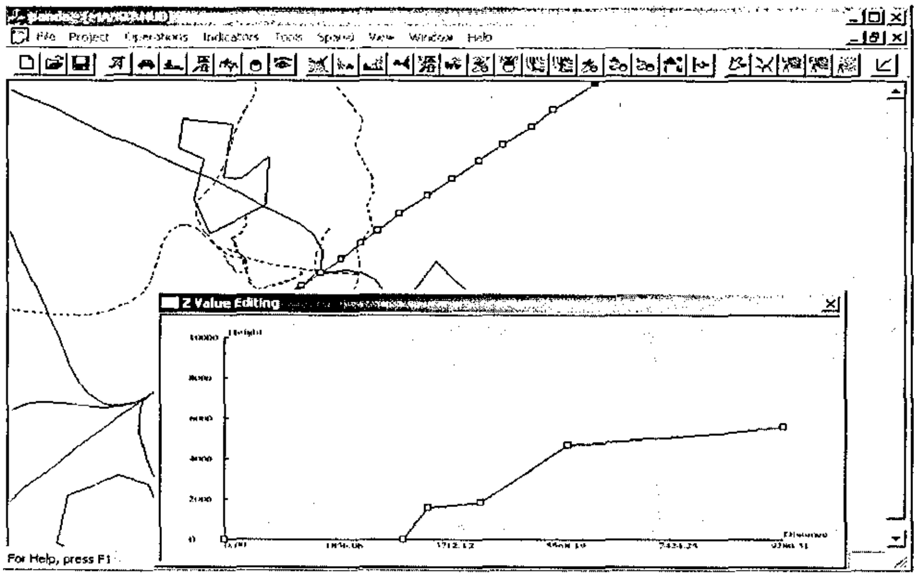


Figure 6.19: 3D design facility in PANDA

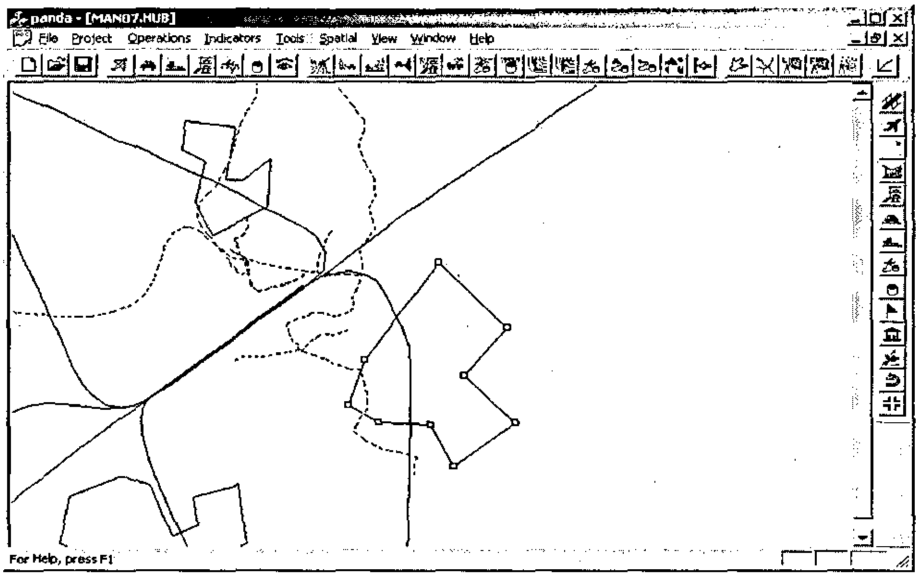

panda - [MAN07.HUB]
File Project Operations Indicators Tools Spatial View Window Help
For Help, press F1

Figure 6.20: Spatial design facility in PANDA

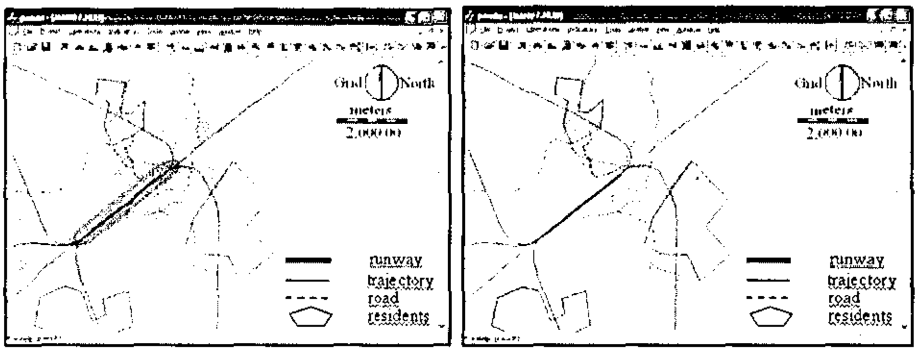


Figure 6.21: Spatial analysis output in PANDA

6.4.5 Sustainability evaluation

One of the most distinctive features of Panda is its ability to provide a sustainability evaluation considering both airport and their residents in the same time. At the moment, most of other systems simply consider sustainability in the view of airport operator only, and they have not taken into account the different perceptions of different people in the vicinity of airports. Such a one sided consideration brings a big gap between the two parties. On airport side, with the development of aircraft technology, they are claiming that they have controlled the noise level at an reasonable level. However, the residents still feel disturbance in their lives, and many of them are against of any expansion of the airport operation. In fact, what the airport has claimed is really true. The noise level has been controlled to some given level. However, the problem is that the given level is not necessarily acceptable to everyone. Because of the difference between human perceptions, an acceptable noise level for A may not be so for B. Therefore, a satisfactory noise distribution in the view of airport operator may be very annoying for some local residents. To satisfy both sides, the acceptable noise distribution has to take the different perceptions into account.

As stated in Chapter 3, we adopt the distance between intuitionistic fuzzy sets to measure the sustainability in Panda. The concept is not only applicable to the airport as a whole, it is also possible to analyse sustainability of each residential area as well. In Panda, we can derive the sustainability of a specific person at a specific location, a group of people in the same residential area and the whole airport. In the case of an individual, the sustainability is calculated as a distance between two sets with single element. For specific residential area, it could be considered as a subset of the airport. Hence, the sustainability could be revealed not only for the airport, but also for individuals or different residential areas as well. Figure 6.22 and 6.23 demonstrate the result for noise sustainability analysis for the residential areas. Figure 6.24 and 6.25 give the comparison of the changes of noise level and noise probability over a given noise level. Obviously, the maximum noise level itself cannot reflect the disturbance increase revealed by the probability model. It illustrates the efficiency of the proposed fuzzy distance and probability model.

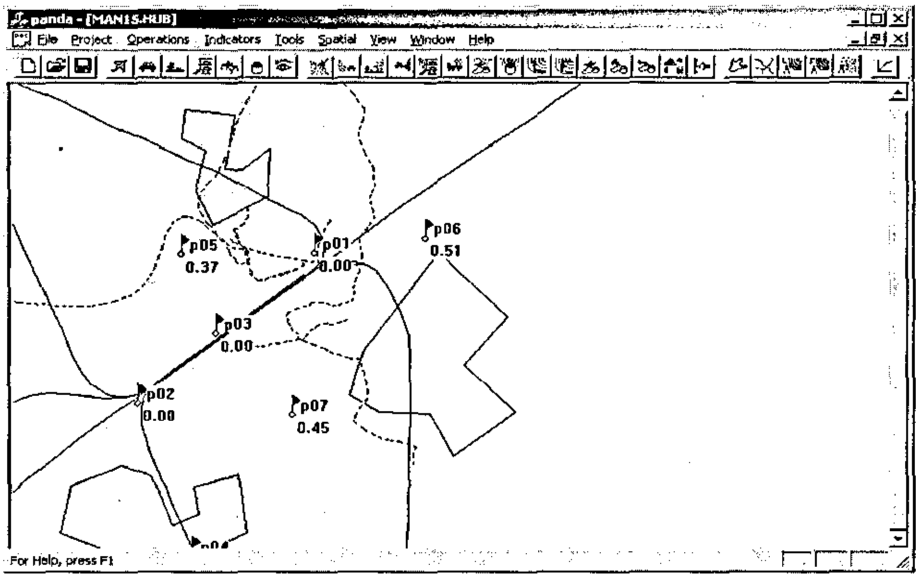


Figure 6.22: Noise sustainability at specific location for a specific resident

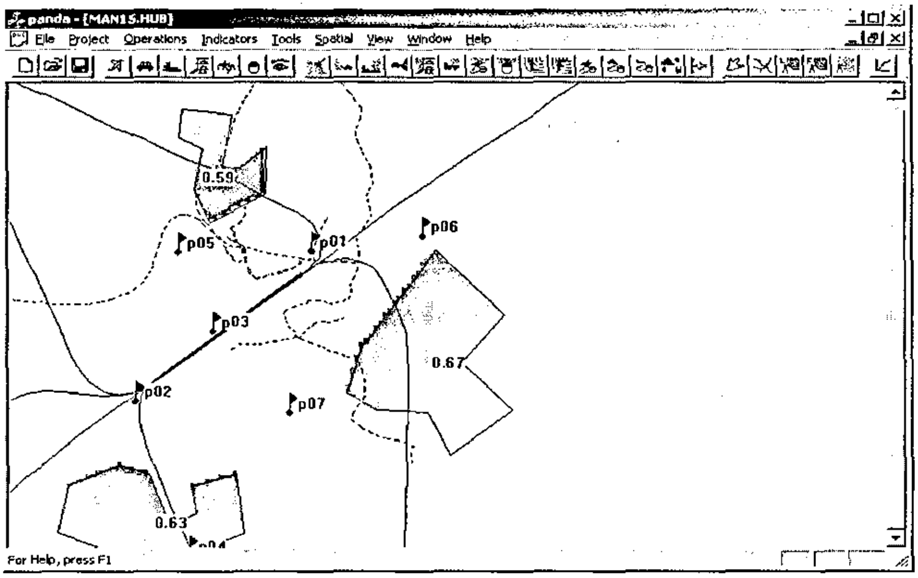


Figure 6.23: Noise sustainability at specific area for a specific community

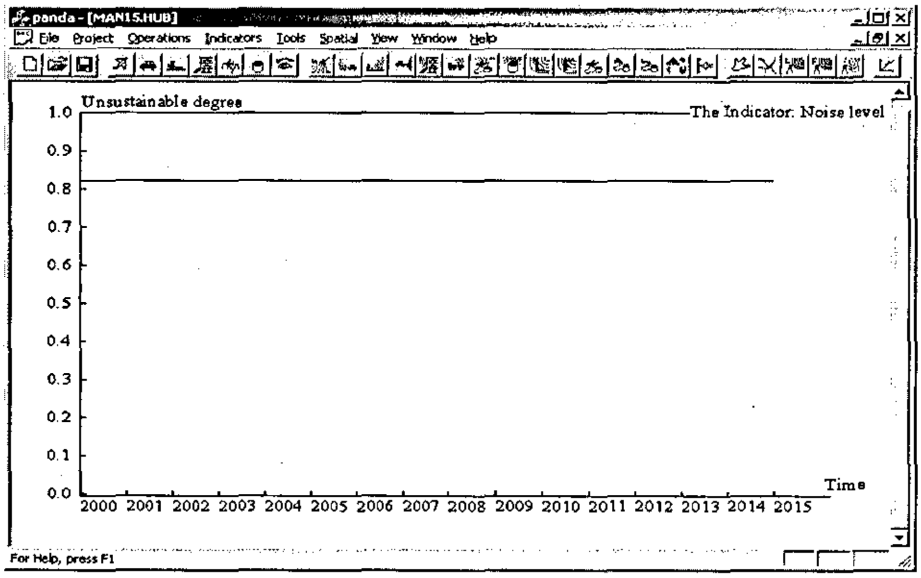


Figure 6.24: The maximum noise level does not increase even if the operation increases

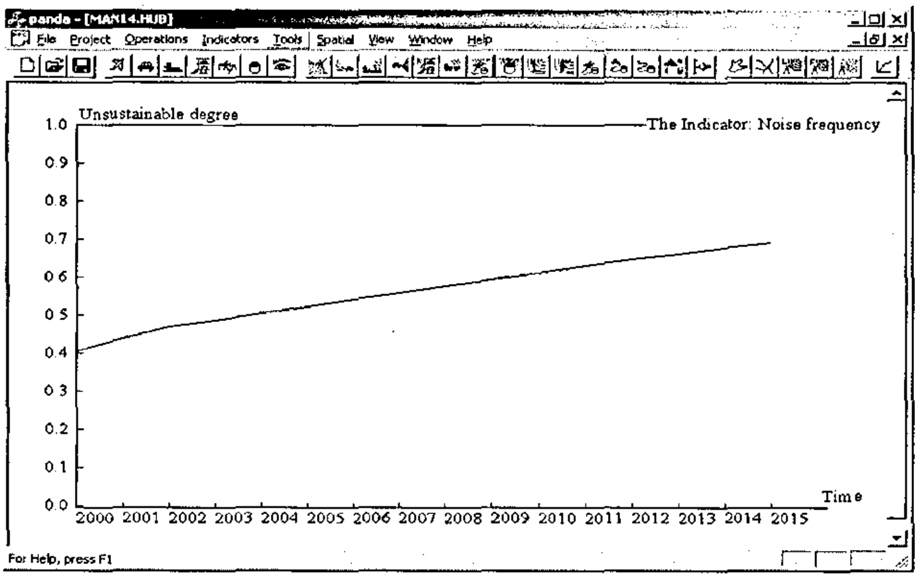


Figure 6.25: Noise frequency indicates that noise disturbance increases with operation

Chapter 7

Conclusions

Airport environment is influenced by many different factors and uncertainty is a significant part of the system. Its uncertainties come from different sources, such as weather change, human perception, operation parameters etc. Therefore, a system for airport environment evaluation has to build up sufficient mechanism in dealing with the uncertainties involved in the operation of airports. This thesis has investigated the applicability of soft computing technology into airport environment evaluation systems, and focused on the technology of neural networks, fuzzy sets, rough sets and grey sets. Under the concept of sustainable development of airports, new methods are defined in establishing, training and interpreting a neural network, and the uncertainty representation of GIS object using grey sets and rough sets as well as the sustainability evaluation using interval-valued/intuitionistic fuzzy sets. With a set of available noise data from Manchester airport, some of our proposed models were verified and the result is very promising. Based on all these works a prototype system was developed for airport environment evaluation – Panda.

It is clear that uncertainty due to the airport operation has to be tackled in an airport environment evaluation system due to its significance. Considering the existence of uncertainties, 3D Hausdorff distances and spherical distances between intuitionistic fuzzy sets are defined and applied to derive the noise sustainability of an airport's operation. Uncertainty propagation is investigated, and it is proved that there are bounds in rough sets operation. To unify fuzzy sets, rough sets and grey systems, grey sets are defined and their relationship with fuzzy sets and rough sets are investigated. Based on grey sets, grey geometry is proposed to represent uncertainties in Geographical Information Systems. In addition to uncertainty, the unknown relationship between airport input and its waste output is another difficulty, and neural networks are adopted to address this problem. To improve the neural network training and explanation, new parameters and structure are defined and applied to network evaluation and data mining. RSE and GRSE are effective parameters in establishing the properties of neural network models. Together with the proposed dynamic state space, RSE and GRSE help with interpretation of results from a trained neural network. PRSE and GPRSE are useful in evaluating a trained neural network. Furthermore, it is shown that a redundant structure is beneficial to neural network training and compound eye inputs are effective in network training as well. The noise simulation through in-situ data proves that our model has better adaptability to local conditions than standard models like INM. Noise disturbance can be better tackled with a probability distribution, which can reveal those increased disturbances hidden by maximum noise level measurements. These ideas have been implemented in the prototype PANDA and NRSE with a 3D facility to carry out WHAT-IF scenarios and would be a useful addition to commercial systems if adopted. To simulate social intelligence in human society, a social intelligence decision support framework is also proposed which has a pyramid style architecture.

The research work has demonstrated the usefulness of soft computing in airport environment decision support systems. The proposed application of intuitionistic fuzzy sets and grey geometry are especially promising for future applications. However, there are still limitations to the present system. The present work does not take multiple indicators into consideration in the measurement of sustainability in the form of distance between intuitionistic fuzzy sets, but multiple indicators are inevitable in real world situation if the whole airport needs to be evaluated. The proposed pyramid architecture has not been fully implemented in the prototype, and some human involvement is still needed. The next step for the proposed system is to implement a negotiating and compromising model among different members in a committee using soft computing technology. Although spherical distance between intuitionistic fuzzy sets is defined, its application still needs exploring thoroughly. Noise is the only indicator being investigated in details here, and other relevant indicators need to be investigated as well. Grey geometry provides a new way in representing spatial uncertainty associated with spatial operations, and it opens a new field in the research of spatial uncertainty models. A further step is to link it with temporal and other uncertainties. The coupled effect of spatial, temporal, operational and perceptional uncertainties is an interesting area for further research.

Appendix A

NRSE objects

A.1 Data objects

A.1.1 CNetNodes

```
class CNetNodes : public CObject
   {
   protected:
   DECLARE_SERIAL(CNetNodes)

   public:
   int m_numNewPara;
   int m_FunctionType;
   double m_RseOpt;
   CNetNodes(CNetNodes& node);
   CNetNodes();
   virtual C̃NetNodes();
   double m_NodeValue, m_NodeError;
   CNetNodes &operator=( CNetNodes& );
   virtual void Serialize(CArchive& ar);
   };
```

A.1.2 CNetParameters

```
class CNetParameters : public CObject
   {
   public:
   CNetParameters();
   virtual C̃NetParameters();
   protected:
   DECLARE_SERIAL(CNetParameters)

   public:
   int* m_NodeNumArray;
   double m_ErrorLimit, m_Error, m_InitError;
   double m_LearnStep, m_LearnMomentum;
   UINT m_Step, m_StepLimit;
   UINT m_StepEnd, m_InitStep, m_NumSample;
   int m_LayerNum, m_StepDraw;
   CArray<CNetWeights,CNetWeights> m_WeightArray;
   CArray<CNetNodes,CNetNodes> m_NodeArray;
   double** m_SampleIn;
   double** m_SampleOut;
   double* m_SampleError;
   CInputOutputs* m_Input;
   CInputOutputs* m_Output;
   CViewEnvironment m_EnView;
   double m_ModifyE;

   public:
   long m_ValidationIndex;
   int m_Validation;
   void ClearSamples();
   double m_DynamicErrorLimit;
   int m_numNewPara;
   int m_RateCompound;
   CString m_Name;
   void ReInitParameters();
   virtual void Serialize(CArchive& ar);
   };
```

A.1.3 CNetWeights

```
class CNetWeights : public CObject
   {
   protected:
   DECLARE_SERIAL(CNetWeights)

   public:
    int m_numNewPara;
    BOOL m_CanModify;
    CNetWeights(CNetWeights& weight);
    CNetWeights();
    virtual C̃NetWeights();
    CNetWeights &operator=( CNetWeights& );
    double m_Weight, m_OldWeight, m_DetWeight, m_OldDetWeight;
    virtual void Serialize(CArchive& ar);
    };
```

A.1.4 CSample

```
class CSample : public CObject
    {
    public:
    CSample();
    virtual C̃Sample();

    protected:
    DECLARE_SERIAL(CSample)

    public:
    CStringArray m_SamInput;
    CStringArray m_SamOutput;
    CSample &operator=( CSample& );

    public:
    BOOL Compare(CSample& s);
    CSample(CSample& c);
    void Copy(CSample& s);
    virtual void Serialize(CArchive& ar);
    };
```

A.1.5 CInputOutputs

```
class CInputOutputs : public CObject
    {
    protected:
    DECLARE_SERIAL(CInputOutputs)

    public:
```

```
    CInputOutputs();
    virtual C̃InputOutputs();
    CString m_Name;
    BOOL m_IsConcept;
    double m_Max, m_Min;
    CStringArray m_ConceptRule;
    double* m_ConceptValue;
    virtual void Serialize(CArchive& ar);
    };
```

A.2 Management objects

A.2.1 CNRSEDoc

```
class CNRSEDoc : public COleServerDoc
    protected:
    CNRSEDoc();
    DECLARE_DYNCREATE(CNRSEDoc)

    public:
    CNRSESrvrItem* GetEmbeddedItem()
    return (CNRSESrvrItem*)COleServerDoc::GetEmbeddedItem();
    virtual BOOL OnNewDocument();
    virtual void Serialize(CArchive& ar);

    protected:
    virtual COleServerItem* OnGetEmbeddedItem();
    unsigned m_StartTime;
    CGISMapDoc ReadGISMapDoc(char* lpszPathName);
    CStringArray AutoInput;
    CSample InterToOut(UINT i, BOOL IsReason);
    void ReasonRse(int);
    void ReasonBP();
    void ReInitRse();
    void GetRse(int type, int nOut);
    BOOL ReasonInput();
    double DoBpLearn(UINT m_CurSamNo, BOOL IsLearn);
    void BpAdjustWeight();
    void BackPropError();
    double GetError(UINT m_CurSamNo);
    void BpForwardReason(BOOL IsRse);
    BOOL IsInit;
    CNetParameters m_NetPara;
    virtual C̃NRSEDoc();
    int WeightNumber(int layer, int i, int j);
    int NodeNumber(int a, int b);
    void SampleClear();
    #ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
    #endif

    protected:
    virtual CDocObjectServer* GetDocObjectServer(LPOLEDOCUMENTSITE
pDocSite);
    afx_msg void OnFileNew();
    afx_msg void OnFileNewSample();
    afx_msg void OnFileOpenSample();
    afx_msg void OnFileAddFilesample();
    afx_msg void OnFileAddHandsample();
    afx_msg void OnEditWeights();
    afx_msg void OnFileNewFromdb();
    afx_msg void OnReasondb();
    afx_msg void OnReasongismap();
    afx_msg void OnEditObservation();
    afx_msg void OnGisMapann();
    afx_msg void OnGisSaveAnnbmp();
    afx_msg void OnGisOpenAnnbmp();
    afx_msg void OnFileExportSampleDb();
    afx_msg void OnFileExportAnnmodel();
    afx_msg void OnFileOpenAppann();
    afx_msg void OnReasondbRse();
    DECLARE_MESSAGE_MAP()
    afx_msg BSTR OutputValue(short Number);
    afx_msg void ExcuteBpReason();
    afx_msg void ExcuteRseReasoning();
    afx_msg BOOL LoadExistedNet(LPCTSTR NetName);
    afx_msg BOOL InputValue(short Number, LPCTSTR Value);
    DECLARE_DISPATCH_MAP()
    DECLARE_INTERFACE_MAP()

    private:
    void ScaleRse(int layer);
    void InitWeight();
    void DoTransfer(CStringArray& s, double* v, BOOL IsIn, float noise);
    };
```

A.2.2 CNRSEView

```
class CNRSEView : public CScrollView {
    protected:
    CNRSEView();
    DECLARE_DYNCREATE(CNRSEView)

    public:
    CNRSEDoc* GetDocument();
    CNRSECntrItem* m_pSelection;
    virtual void OnDraw(CDC* pDC);
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);

    protected:
    virtual void OnInitialUpdate();
    virtual BOOL OnPreparePrinting(CPrintInfo* pInfo);
    virtual void OnBeginPrinting(CDC* pDC, CPrintInfo* pInfo);
    virtual void OnEndPrinting(CDC* pDC, CPrintInfo* pInfo);
    virtual BOOL IsSelected(const CObject* pDocItem) const;
    virtual C̃NRSEView();
    #ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
    #endif

    protected:
    afx_msg void OnContextMenu(CWnd*, CPoint point);
    afx_msg void OnDestroy();
    afx_msg void OnSetFocus(CWnd* pOldWnd);
    afx_msg void OnSize(UINT nType, int cx, int cy);
    afx_msg void OnInsertObject();
    afx_msg void OnCancelEditCntr();
afx_msg void OnCancelEditSrvr();
afx_msg void OnViewStructure();
afx_msg void OnViewSamples();
afx_msg void OnViewWeight();
afx_msg void OnViewSampleErrors();
afx_msg void OnReasonBp();
afx_msg void OnLearnBp();
afx_msg void OnLearnStop();
afx_msg void OnViewFigure();
afx_msg void OnLearnOn();
afx_msg void OnUpdateLearnOn(CCmdUI* pCmdUI);
afx_msg void OnUpdateLearnStop(CCmdUI* pCmdUI);
afx_msg void OnUpdateLearnBp(CCmdUI* pCmdUI);
afx_msg void OnReasonRse();
afx_msg void OnViewGrse();
afx_msg void OnViewThreshold();
afx_msg void OnViewGfrse();
afx_msg void OnViewName();
afx_msg void OnViewCurrentgis();
afx_msg void OnRButtonDown(UINT nFlags, CPoint point);
afx_msg void OnGisOpen();
afx_msg void OnGisExtract();
afx_msg void OnGisOpenbmp();
afx_msg void OnGisModifybmp();
afx_msg void OnViewCurrentbmp();
afx_msg void OnLButtonDblClk(UINT nFlags, CPoint point);
afx_msg void OnViewNodevalue();
afx_msg void OnReasonPrse();
DECLARE_MESSAGE_MAP()

private:
CSize m_sizeScreen;
COLORREF m_oldColor;
CPoint m_CheckP;
BOOL m_GisOpen;
CGISMapDoc mapinfo;
unsigned char* m_GisValue;
CSize DrawGisMap(CDC* dc);
UINT DrawThreshold(CDC* dc, int height);
UINT DrawRse(CDC* dc, int height);
```

```
void DoBpLearnLoop(int LearnType, long Index);
UINT DrawFigureCoordinate(CDC* dc, int& dx, int& dy, int vPos);
CSize m_ScaleArray[20];
UINT DrawError(CDC* dc, int height);
UINT DrawBpReasonResult(CDC * dc, int height);
UINT DrawSampleErrors(CDC* dc, int height);
UINT DrawSamples(CDC* dc, int height);
UINT DrawStructure(CDC* pDC, int height);
UINT DrawWeights(CDC* dc, int height);
UINT DrawNodeValue(CDC* dc, int height);

public:
int m_TextHeight;
CDC dcMem;
CImageObject *m_pImageBmp;
void ReInitVariables();
};
```

Appendix B

Objects for CAD facility

B.1 Objects for points

B.1.1 CRealPoint

```
class CRealPoint : public CObject
    {
    protected:
    DECLARE_SERIAL(CRealPoint)

    public:
    double m_Z;
    double m_Y;
    double m_X;
    double m_M;
    int m_Sign;
    double GetRealDistance(CRealPoint, CMapScale, int);
    CRealPoint GetExtendPoint(CRealPoint, double);
    void RealToScreen(CMapScale&);
    void ScreenToReal(CMapScale&);
    double GetShortDisToSegment(CRealPoint, CRealPoint, int);
    double GetDisPtoLine(CRealPoint, CRealPoint, int);
    double GetDistance(CRealPoint&, int);
    CRealPoint ToInt();
    void Rotate(double, double, double);
    BOOL operator!=(const CRealPoint&);
    BOOL operator==(const CRealPoint&);
    CRealPoint();
    CRealPoint(double m_X, double m_Y, double m_Z=0, double m_M=0, int sign=0);
    CRealPoint(CRealPoint&);
    virtual C̃RealPoint();
    CRealPoint &operator=(const CRealPoint& );
    virtual void Serialize(CArchive& ar);
    };
```

B.1.2 CJoint

```
class CJoint
    {
    public:
    int m_Kind;
    int m_Position;
    CJoint& operator=(CJoint&);
    int m_Seg2;
    int m_Seg1;
    int m_IdSub2;
    int m_IdSub1;
    int m_IdEntity2;
    int m_IdEntity1;
    CRealPoint m_Joint;
    CJoint();
    CJoint(CRealPoint, int, int, int, int, int, int);
    virtual C̃Joint();
    };
```

B.2 Object for entity

```
class CEntity : public CObject
    {
    protected:
    double GetBoundaryRegn(CMapScale&, int);
    int CheckOverlayUncertainty(CEntity&, CMapScale&, int);
    DECLARE_SERIAL(CEntity)

    public:
    CArray<CRealPoint, CRealPoint> m_PointsSet;
    CArray<int, int> m_StyleSet;
    int m_Sort;
    double m_Capacity;
    CString m_Statement;
    long m_Duration;
    long m_StartTime;
    CArray<double,double> m_SegFeature; //Not complete yet, reserved for
future use !
    COLORREF m_BackGroundColour;
    CArray<double,double> m_Uncertainty;
    CStringArray m_Variable; //Not complete yet, reserved for future use !
    CString m_Function;
    CString m_Name;
    BOOL m_Visible;
    BOOL m_Closed;
    int m_FillStyle;
    int m_LineStyle;
    int m_LineWidth;
    CTypedPtrList<CPtrList,CEntity*> m_SubEntity;
    COLORREF m_FillColour;
    COLORREF m_LineColour;

    void AssignUncertainty(double);
    double GetUncertainty(CMapScale& scale);
    CRealPoint GetLabelPos();
    double GetRealDistance(CRealPoint, CRealPoint, CMapScale, int);
    BOOL GetDistributionPoints(double, CArray<CRealPoint,CRealPoint>&,
CMapScale);
    double FindEndValue(int, BOOL);
    void SetLineWidth(int);
    void SetLineStyle(int);
    void SetLineColour(COLORREF);
    double GetArea(CMapScale&, int sub=-1);
    void Transform3DLine(CMapScale&);
    CRealPoint GetNodeCoordinate(int,int sub=0);
    void DrawColourMark(CDC*, int, int, COLORREF);
    int SelectNode(CPoint, int& sub);
    int GetNodeNumber(int sub=0);
    double SetZValue(double, int, int sub=0, BOOL IsTurnPoint=FALSE);
    double GetZValue(int, int sub=0);
    double GetRouteDistance(int, int, CMapScale&, int sub=0);
    double GetLength(CMapScale&, int sub=0);
    void DoScale(double,double,double);
    BOOL EntityInterchange(CEntity&);
    void RemoveRepeatBoundarys();
    int LineInEntity(CRealPoint, CRealPoint);
    int FindCommonPoint(CArray<CJoint,CJoint>&);
    int RemoveFalseRing(CTypedPtrList<CPtrList, CEntity*>&, double);
    int RemoveRepeats();
    void VectorMapOverlay(CTypedPtrList<CPtrList, CEntity*>*, CTypedPtrList<CPtrList, CEntity*>*, int, CTypedPtrList<CPtrList, CEntity*>*, double, CMapScale&);
    BOOL operator!=(CEntity&);
    BOOL operator==(CEntity&);
    void TransformToLine();
    void MatchBoundary(CRealPoint, int&, int&);
    int PtStrictInEntity(CRealPoint);
    double GetDisPtoLine(CRealPoint, CRealPoint, CRealPoint);
    BOOL GetInterPoint(CRealPoint&, double);
    BOOL CheckRepeatJoint(CRealPoint, CArray<CJoint, CJoint>&);
    CEntity* FindEntity(int);
    void MatchPoints(CArray<CJoint, CJoint>&, BOOL);
    BOOL ClearRepeatJoint(CRealPoint, CArray<CJoint, CJoint>&);
    BOOL CheckRepeatPoint(CRealPoint, CArray<CRealPoint, CRealPoint>&, double);
    int GetJointSLPt(CRealPoint, CRealPoint, CRealPoint, CRealPoint, CArray<CJoint, CJoint>&);
    int GetJointLinePoint(CRealPoint, CRealPoint,CArray<CJoint, CJoint>&);
    int GetJointLinePoints(CEntity&, CArray<CJoint, CJoint>&);
    BOOL EntityInclude(CEntity&);
    BOOL FormRing(CTypedPtrList<CPtrList,CEntity*>&, BOOL);
    BOOL IsSelLine(CRealPoint, CRealPoint, CRealPoint, double);
    int InsertNode(CRealPoint, double, double, int);
    CRealPoint GetPointInLine(CRealPoint, CRealPoint, double);
    int PtInEntity(CRealPoint);
    double GetDistance(CRealPoint, CRealPoint, int);
    int OverlayOperation(CEntity&, CTypedPtrList<CPtrList, CEntity*>&, int, double);
    BOOL StretchEllipse(const CEntity&, CPoint, int);
    BOOL StretchPartEllipse(CEntity&, double, double, double, double, double, double);
    CEntity* FindSubEntity(int);
```

```
void Stretch(CEntity&, CPoint&, CSelIndicator&, double, double, double, double, double, double, double, double);
BOOL UpdateMoveXY(CEntity&, double, double, double ,double, int, CSelIndicator&);
void MoveWithSel(CDC*, CPoint&, CPoint, double&, CPoint&, RECT&, BOOL & m_BeginMove, CSelIndicator&, CEntity&);
BOOL MoveWithArcEnds(CDC*, int, CPoint&, CPoint, CSelIndicator&, double&);
int GetRgnEntity(CRgn&, double);
void ReInitializeSub();
BOOL GetExPEllipse(CEntity&, double,double,double,CRealPoint&,double&,int&);
BOOL GetJointPforEllipse(CEntity&, CArray<CJoint,CJoint>&);
BOOL GetBezierJoint(CEntity&, CArray<CJoint,CJoint>&);
BOOL GetJointPforTwoBezier(CEntity&, double, double, double,CRealPoint&, double&, int&);
BOOL GetJointPforBeizer(CEntity&, double, double, double, CRealPoint&, double&);
void AdjustRectCordinates(double,double,double,double,double&,double&,double&,double&);
BOOL FindPInterLine(CEntity&, CArray<CJoint,CJoint>&, int);
BOOL PinArc(CRealPoint, double);
int FindLineCircleInterP(CEntity&, CRealPoint*);
CRealPoint FindBezierPoint(double);
int FindLineInterP(CRealPoint, CRealPoint, CRealPoint, CRealPoint, CRealPoint&, int);
BOOL GetPartCompound(CRealPoint*, int);
int GetIntersection(CEntity&, CArray<CJoint,CJoint>&, int);
BOOL GetCompoundEnds(CRealPoint&, CRealPoint&);
RECT FindStretchCompoundXY();
BOOL StretchPartArc(const CEntity&, CRealPoint, CRealPoint, CRealPoint, int, int);
CPoint UpdateStretchPoint(double, double, CRealPoint, CPoint);
BOOL UpdateStretchXY(const CEntity&, CRealPoint, double, double, int);
void EllipseToBezier();
BOOL StretchArc(const CEntity&, CPoint, int);
CPoint RotatePoint(CPoint, double, double, double);
CRealPoint RotatePoint(CRealPoint, double, double, double);
BOOL GetControlPoints(CPoint *);
CPoint FindCenter();
void ModifySort();
BOOL ShowEntityXYData();
```

```
RECT FindMaxPoints(CPoint);
BOOL DrawCompCtrl(CDC*);
BOOL ReversePointOrder();
CSelIndicator IsSelCurve(double, double, double, double, int, CPoint);
BOOL DrawCompoundPath(CDC*, BOOL);
CSelIndicator IsSelCurve(double, double, int, CPoint, BOOL);
CSelIndicator CheckCompound(int, int, int, long&, CPoint);
BOOL FindPathEnds(double&, double&, double&, double&);
BOOL Rotate(double, double, double);
BOOL Move(double,double);
BOOL FindPartArcEnds(int&, int&, int&, int&, int);
BOOL FindArcEnds9(int&, int&, int&, int&);
BOOL DrawArc(CDC*, int, int, double, double, double, double, double);
BOOL DrawEllipse(CDC*, int, int, double, double, double);
BOOL DrawArcTo(CDC * dc, int, int, double, double, double, double, double);
void DrawMarkForCurve(CDC*, int);
CSelIndicator IsSelCurve(int, CPoint);
void FindArcEnds(int&,int&,int&,int&);
BOOL IsSelControlHandle(int,int, CPoint p);
BOOL IsSelControlHandle(CPoint, CPoint);
void DrawCircleMark(CDC*, int, int);
void DrawFrame(CDC*);
double NearestDis(CPoint, int&);
void DrawSelControl(CDC*);
void Copy(CEntity&);
int IsControlPoint(CPoint);
CSelIndicator CheckLine(CPoint&, int, int, int, int);
CSelIndicator CheckLine(CPoint&, int);
BOOL IsSelPoint(CPoint, CPoint);
RECT FindMaxMiniXY();
int IsSelLine(CPoint, int, int, int, int);
void ReInitializeData();
int IsSelLine(CPoint,int);
CSelIndicator IsSelected(CPoint);
void AdjustRectCordinates(int,int,int,int,int&,int&,int&,int&);
void DrawMark(CDC*, int, int);
void DrawSelMark(CDC*);
void DrawEntity(CDC*, BOOL);
CEntity& operator=(CEntity&);
```

```
CEntity(CEntity&);
virtual void Serialize(CArchive& ar);
CEntity();
virtual C̃Entity();

private:
int RemoveOverlay(CEntity&, CTypedPtrList<CPtrList, CEntity*>&, double);
int SubOverlay(CEntity&, CTypedPtrList<CPtrList, CEntity*>&, double);
void RateEntity(double);
int OverlayWithoutJoint(CEntity&, CTypedPtrList<CPtrList, CEntity*>&, int);
int PolygonOverlay(CEntity&, CTypedPtrList<CPtrList, CEntity*>&, int);
int RingOverlay(CEntity&, CTypedPtrList<CPtrList, CEntity*>&, int, double);
int AdditionOverlay(CEntity&, CTypedPtrList<CPtrList, CEntity*>&, double);
};
```

Appendix C

PANDA

C.1 Data objects

C.1.1 CAircraftEmissionLinkData

```
class CAircraftEmissionLinkData : public CObject
    {
    protected:
    DECLARE_SERIAL(CAircraftEmissionLinkData)

    public:
    CString m_SVFuel;
    CString m_SVclass;
    CString m_SVfreight;
    CString m_SVnumPas;
    CString m_SVvField;
    CString m_SVtbName;
    CString m_SVdbName;
    CString m_EVfreight;
    CString m_EVnumPas;
    CStringArray m_fieldsEFCarparkV;
    CString m_idEFCarparkV;
    CString m_tbEFCarparkV;
    CString m_dbEFCarparkV;
    CString m_NoiseCurveType;
    CString m_thrustNoise;
    CString m_EVNoiseId;
    CString m_typeNoise;
    CStringArray m_disNoise;
    CStringArray m_fieldsNoise;
    CString m_idNoise;
    CString m_tbNoise;
    CString m_dbNoise;
    CStringArray m_fieldsEFMotorway;
    CString m_idEFMotorway;
    CString m_tbEFMotorway;
    CString m_dbEFMotorway;
    CStringArray m_fieldsEFColdStartV;
    CString m_idEFColdStartV;
    CString m_tbEFColdStartV;
    CString m_dbEFColdStartV;
    CStringArray m_fieldsEFRoadInAirport;
    CString m_idEFRoadInAirport;
    CString m_tbEFRoadInAirport;
    CString m_dbEFRoadInAirport;
    CAircraftEmissionLinkData& operator =(const CAircraftEmissionLinkData
&);
    BOOL operator ==(const CAircraftEmissionLinkData &);
    BOOL CheckData();
    BOOL CheckDatabase(CString&, CString, CStringArray&);
    virtual void Serialize(CArchive& ar);
    void ReInitializeData();
    CStringArray m_EFfields;
    CString m_EFidField;
    CString m_EFtbName;
    CString m_EFdbName;
    CStringArray m_FuelCFields;
    CString m_FuelCidField;
    CString m_FuelCtbName;
    CString m_FuelCdbName;
    CStringArray m_Stage;
    CStringArray m_StageValue;
    CStringArray m_Indicators;
    CString m_EVnField;
    CString m_EVeField;
    CString m_EVvField;
    CString m_EVtbName;
    CString m_EVdbName;
    CAircraftEmissionLinkData();
    virtual CAircraftEmissionLinkData();
    };
```

C.1.2 CAircraftGenOptData

```
class CAircraftGenOptData : public CObject
    {
    protected:
    DECLARE_SERIAL(CAircraftGenOptData)

    public:
    void ClearRailData();
    void ClearRoadData();
    CStringArray& GetSurVehicleType(int);
    CAircraftOperationData GetSurVehicledata(int, int, int);
    BOOL GetUnitPasRate(int, CArray<double, double>&);
    CArray<CPassengerNumber,CPassengerNumber> m_PassengerNumber;
    double m_PeakHourRatio;
    double m_WorkHours;
    int m_TimeType;
    CAircraftGenOptData& operator =(const CAircraftGenOptData &);
    BOOL operator ==(const CAircraftGenOptData &);
    virtual void Serialize(CArchive& ar);
    long GetUnitMovement(int);
    BOOL GetUnitValue(int, CArray<double,double>&);
    BOOL GetUnitType(int, CStringArray&);
    int GetTypes(CStringArray&);
    void ReInitialize();
    int m_Type;
    int m_Period;
    int m_First;
    CTypedPtrList<CPtrList, CAircraftOperationData*> m_Units;
    CTypedPtrList<CPtrList, CAircraftOperationData*> m_RoadCar;
    CTypedPtrList<CPtrList, CAircraftOperationData*> m_RoadBus;
    CTypedPtrList<CPtrList, CAircraftOperationData*> m_Train;
    CTypedPtrList<CPtrList, CAircraftOperationData*> m_RoadTruck;
    CTypedPtrList<CPtrList, CAircraftOperationData*> m_MotorwayCar;
    CTypedPtrList<CPtrList, CAircraftOperationData*> m_MotorwayBus;
    CTypedPtrList<CPtrList, CAircraftOperationData*> m_MotorwayTruck;
```

```
   CTypedPtrList<CPtrList, CAircraftOperationData*> m_CarparkCar;
   CTypedPtrList<CPtrList, CAircraftOperationData*> m_CarparkBus;
   CTypedPtrList<CPtrList, CAircraftOperationData*> m_CarparkTruck;
   CAircraftGenOptData();
   virtual C̃AircraftGenOptData();
   };
```

C.1.3 CAircraftOperationData

```
class CAircraftOperationData : public CObject
   {
   protected:
   DECLARE_SERIAL(CAircraftOperationData)

   public:
   void ReInitializeData();
   CStringArray m_VPasRate;
   BOOL operator ==(const CAircraftOperationData &);
   virtual void Serialize(CArchive& ar);
   CAircraftOperationData(CAircraftOperationData&);
   CStringArray m_VRate;
   CStringArray m_VType;
   CAircraftOperationData();
   virtual  CAircraftOperationData();
   CAircraftOperationData& operator =(const CAircraftOperationData&);
   };
```

C.1.4 CFugitiveData

```
class CFugitiveData : public CObject
   {
   protected:
   DECLARE_SERIAL(CFugitiveData)

   public:
   virtual void Serialize(CArchive& ar);
   BOOL operator ==(const CFugitiveData &);
   CFugitiveData& operator =(const CFugitiveData &);
   CFugitiveData(const CFugitiveData&);
   float m_VapourPressure;
   int m_TurnoverNumber;
```

```
    float m_TurnoverFactor;
    int m_TankTime;
    int m_TankNumber;
    CString m_TankName;
    float m_TankCapacity;
    float m_ProductFactor;
    float m_MolecularWeight;
    float m_Diameter;
    float m_AverageVapourHeight;
    float m_AverageTemperatureChange;
    float m_AveragePressure;
    CFugitiveData();
    virtual  CFugitiveData();
    };
```

C.1.5 CIndicatorLimits

```
class CIndicatorLimits : public CObject
    {
    protected:
    DECLARE_SERIAL(CIndicatorLimits)

    public:
    double m_RefDistance;
    double m_FuelConsumption;
    CIndicatorLimits(const CIndicatorLimits&);
    BOOL operator==(const CIndicatorLimits&);
    CIndicatorLimits& operator=(const CIndicatorLimits&);
    double m_WaterLimits;
    double m_ElectricityLimits;
    double m_WasteLimits;
    void ReInitialiseData();
    virtual void Serialize(CArchive& ar);
    double m_NoiseTimeLimits;
    double m_NoiseLevelLimits;
    CArray<double, double> m_EmissionDispersionLimits;
    CArray<double, double> m_EmissionLoadingLimits;
    CIndicatorLimits();
    virtual C̃IndicatorLimits();
    };
```

C.1.6 CInfrustructure

```
class CInfrustructure : public CObject { protected:
    DECLARE_SERIAL(CInfrustructure)

    public:
    CString m_Function;
    void ReInitializeData();
    virtual void Serialize(CArchive& ar);
    BOOL operator !=(const CInfrustructure &);
    BOOL operator ==(const CInfrustructure &);
    CInfrustructure& operator =(const CInfrustructure &);
    CInfrustructure(const CInfrustructure&);
    double m_Volume;
    long m_Capacity;
    long m_ServicePeriod;
    long m_StartTime;
    CString m_Name;
    CInfrustructure();
    virtual C̃Infrustructure();
    };
```

C.1.7 CNoiseData

```
class CNoiseData : public CObject
    {
    protected:
    DECLARE_SERIAL(CNoiseData)

    public:
    CString m_PositionField;
    CString m_tbName;
    CString m_dbName;
    void ReInitialize();
    CString m_OpTypeField;
    CString m_ArcTypeField;
    CString m_TimeField;
    CString m_DisField;
    CString m_LevelField;
    CNoiseData(const CNoiseData&);
    virtual void Serialize(CArchive& ar);
```

```
   CNoiseData();
   virtual C̃NoiseData();
   CNoiseData& operator=(const CNoiseData&);
   BOOL operator==(const CNoiseData&);
   };
```

C.1.8 CPassengerNumber

```
class CPassengerNumber : public CObject
   {
   protected:
   DECLARE_SERIAL(CPassengerNumber)

   public:
   double m_PasBusRate;
   double m_PasCarRate;
   double m_EmpBusRate;
   double m_EmpCarRate;
   virtual void Serialize(CArchive& ar);
   BOOL operator !=(const CPassengerNumber&);
   BOOL operator ==(const CPassengerNumber&);
   CPassengerNumber& operator =(const CPassengerNumber&);
   CPassengerNumber(const CPassengerNumber&);
   long m_ExtraSurPas;
   long m_Employee;
   long m_AirMovement;
   CPassengerNumber();
   virtual C̃PassengerNumber();
   };
```

C.1.9 CRunwayData

```
class CRunwayData : public CObject
   {
   protected:
   DECLARE_SERIAL(CRunwayData)

   public:
   CString m_Duration;
   CString m_StartTime;
   CRunwayData& operator =(const CRunwayData &);
```

```
    BOOL operator ==(const CRunwayData &);
    CStringArray m_Fields;
    CString m_IdField;
    CString m_tbName;
    CString m_dbName;
    void ReInitializeData();
    virtual void Serialize(CArchive& ar);
    CStringArray m_StageValue;
    CString m_Length;
    CString m_Capacity;
    CString m_Usage;
    CString m_Name;
    CRunwayData();
    virtual C̃RunwayData();
    };
```

C.1.10 CServiceGseData

```
class CServiceGseData : public CObject
    {
    protected:
    DECLARE_SERIAL(CServiceGseData)

    public:
    void ReInitializeData();
    CString m_Vehicle;
    CArray<float,float> m_Factors;
    float m_ConsumptionRate;
    virtual void Serialize(CArchive& ar);
    CServiceGseData(const CServiceGseData&);
    float m_RatioAirside;
    float m_ConversionFactor;
    CArray<float,float> m_FuelService;
    CArray<float,float> m_FuelGSE;
    CServiceGseData();
    virtual C̃ServiceGseData();
    CServiceGseData& operator =(const CServiceGseData &);
    BOOL operator ==(const CServiceGseData &);
    };
```

C.1.11 CSurfaceData

```
class CSurfaceData : public CObject
    {
    protected: DECLARE_SERIAL(CSurfaceData)

    public:
    CString m_Vehicle;
    int m_Time;
    CSurfaceData(const CSurfaceData&);
    virtual void Serialize(CArchive& ar);
    double m_Length;
    long m_Number;
    CString m_Name;
    CSurfaceData();
    virtual C̃SurfaceData();
    CSurfaceData& operator =(const CSurfaceData &);
    BOOL operator ==(const CSurfaceData &);
    };
```

C.1.12 CWasteWaterElectricityData

```
class CWasteWaterElectricityData : public CObject
    {
    protected:
    DECLARE_SERIAL(CWasteWaterElectricityData)

    public:
    CString m_AnnFile;
    void ReInitializeData();
    virtual void Serialize(CArchive &ar);
    BOOL operator !=(const CWasteWaterElectricityData & e);
    BOOL operator ==(const CWasteWaterElectricityData & e);
    CWasteWaterElectricityData& operator =(const CWasteWaterElectricityData & e);
    CWasteWaterElectricityData(const CWasteWaterElectricityData&);
    CString m_PlaceField;
    CString m_TimeField;
    CString m_tbName;
    CString m_dbName;
    CString m_ValueField;
    CString m_NumPasField;
    CWasteWaterElectricityData();
    virtual C̃WasteWaterElectricityData();
    };
```

C.1.13 CWeatherData

```
class CWeatherData : public CObject
    {
    protected:
    DECLARE_SERIAL(CWeatherData)

    public:
    int m_AtmosStability;
    void ReInitializeData();
    virtual void Serialize(CArchive& ar);
    CWeatherData(const CWeatherData&);
    double m_WindDirection;
    double m_WindSpeed;
    double m_Temperature;
    CWeatherData();
    virtual C̃WeatherData();
    CWeatherData& operator =(const CWeatherData &);
    BOOL operator ==(const CWeatherData &);
    };
```

C.1.14 CHubAirport

```
class CHubAirport : public CObject
    {
    protected:
    DECLARE_SERIAL(CHubAirport)

    public:
    CString m_ExeInfo;
    double** m_SusIndicatorValueSet;
    CStringArray m_SusIndicatorNameSet;
    CWasteWaterElectricityData m_WasteData;
    CWasteWaterElectricityData m_WaterData;
    CWasteWaterElectricityData m_ElectricityData;
    CWeatherData m_Weather;
```

```
CStringArray m_AnnLevelAircraft;
CStringArray m_AnnLevelFiles;
CStringArray m_AnnProbaAircraft;
CStringArray m_AnnProbaFiles;
CNetParameters m_Ann;
float m_NoiseLimit;
double m_detNoiseDis;
double m_minNoiseDis;
double m_maxNoiseDis;
CArray<double,double> m_NoiseCurve;
CArray<double,double> m_NoiseDis;
CNoiseData m_NoiseData;
CArray<CSurfaceData,CSurfaceData> m_MotorwayData;
CArray<CSurfaceData,CSurfaceData> m_CarParkLinkData;
CArray<CSurfaceData,CSurfaceData> m_CarParkData;
CArray<CSurfaceData,CSurfaceData> m_RailData;
CServiceGseData m_ServiceGseData;
CIndicatorLimits m_IndicatorLimits;
CArray<CFugitiveData,CFugitiveData> m_FugitiveData;
CTypedPtrList<CPtrList,CRunwayData*> m_RunwayData;
CRunwayUsageData m_RunwayUsageData;
CAircraftGenOptData m_AircraftOptData;
CAircraftEmissionLinkData m_EmissionLink;
double*** m_Emission;
CArray<CInfrustructure,CInfrustructure> m_Infrustructure;
void SetMaxDisplayValue(double);
void SetupEmissionArray(int, int, int);
void InsertInfrustructrure(CEntity&, CMapScale&);
void ModifyInfrustructure(CString, CEntity&, CMapScale&);
void DeleteInfrustructure(CString, CString);
CString CheckCapacity();
BOOL GetGeneralNoise(BOOL);
void GetTotalIndicatorValue(double**);
BOOL GetSusIndicators();
BOOL GetSeasonRation(double*);
int GetWasteWaterElectricityConsumption();
BOOL GetSurfVehicleFuelCoef(CStringArray&, CArray<double,double>&);
BOOL GetSurTrafficFuelConsumption(BOOL);
BOOL GetAirFuelConsumption();
double GetInfrusVolume(CString);
```

```
    int GetNumSpecificInfrustructure(CString);
    void ClearSpecificInfrustructure(CString);
    void GetInfrustructure(CString, CArray<CInfrustructure, CInfrustructure>&);
    BOOL GetVehicleClass(CStringArray&, CArray<int,int>&);
    BOOL GetUnitAirFreight(CStringArray&, CArray<double,double>&);
    BOOL GetYearAirFreight(CArray<double,double>&);
    BOOL GetUnitVFreight(CStringArray&, CArray<double,double>&);
    BOOL GetSurVPasNumber(CStringArray&, CArray<int,int>&);
    long GetSurVehicleNumber(int, CString, CString, int);
    BOOL GetAircraftPasNumber(CStringArray&, CArray<long, long>&);
    BOOL GetAirPasYearNumber(CArray<long,long>&);
    BOOL GetVehicleType(CStringArray&, CString);
    BOOL GetCarparkEmissionFactor(CStringArray&, double**);
    BOOL GetCarparkEmission();
    double GetEmissionDispersion(CRealPoint, CRealPoint, double);
    void DoEmissionDispersion(CArray<CRealPoint, CRealPoint>&, CArray<double, double>&, CRealPoint, double);
    BOOL LoadAnn(CString);
    int WeightNumber(int, int, int);
    int NodeNumber(int, int);
    CSample InterToOut();
    void BpForwardReason();
    void DoTransfer(CStringArray&, double*, float);
    BOOL AnnReason(CStringArray&, CSample&);
    BOOL WriteNoiseSamples(CString, CString, CString, int, int);
    BOOL GetMaxMinMonitorNoise(CString, CString, CString, double&, double&);
    BOOL ReadNoiseTypes(BOOL, CStringArray&);
    BOOL DrawNoiseCurve(CDC*, CString, CRect, int);
    BOOL GetMaxMinNoiseMonitorDis(CString, CString, CString, double&,double&);
    BOOL GetMonitorNoiseCurve(CString, CString, CString, double, int);
    double GetNoiseDataStatistics(CString, CString, CString, double, double);
    void DefineIndicatorLimits();
    BOOL GetFugitiveEmission();
    BOOL GetGseEmission();
    BOOL GetAirportServiceEmission();
    int GetRoadStage(int, CStringArray&);
    BOOL GetColdStartEmissionFactor(CStringArray&,double**);
    BOOL GetColdStartEmission();
    void ReInitializeEmission();
    BOOL GetSurEmissionFactors(BOOL,CStringArray&,double**);
    BOOL GetSurTransEmissions(BOOL);
    BOOL GetRunwayUsageDB(CStringArray&, double**);
    BOOL CheckDbData();
    BOOL GetSOperationTime(double**);
    BOOL GetVOperationTime(CStringArray&, double***);
    CRunwayData* GetRunwayData(int);
    virtual void Serialize(CArchive& ar);
    void ReInitializeData();
    BOOL GetFuelConsumption(const CStringArray&, double**);
    BOOL GetEngineType(const CStringArray&, CStringArray&, CArray<int,int>&);
    CPoint DrawFigCoordinate(CDC*, int&, int&, int, int, int,CString, int,
CStringArray* monitor=NULL);
    CPoint DrawIndicators(CDC*, int, int, int, CString, CRect&, int, int, CPoint,
CStringArray* monitor=NULL);
    BOOL GetAircraftEmission();
    CHubAirport();
    virtual C̃HubAirport();
    BOOL GetEmissionFactors(CStringArray&, double**);
    private:
    double m_MaxValue;
    int m_NumEmissionYear;
    int m_NumIndicator;
    int m_NumStage;
    };
```

C.2 Management objects

C.2.1 CPandaDoc

```
class CPandaDoc : public COleServerDoc
    {
    protected:
    CPandaDoc();
    DECLARE_DYNCREATE(CPandaDoc)

    public:
    CString m_dbName;
    BOOL m_dBMark;
    float m_Version;
```

```
int m_idEmission;
CHubAirport m_Airport;
CStringArray m_FieldsType;
CStringArray m_DatabaseFields;
CString m_DocName;
int m_AnalysisType;
CBkgImage m_BkgImage;
CMapScale m_Scale;
CActionSign m_ActionSign;
CTypedPtrList<CPtrList, CEntity*> m_EntitySet;
CTypedPtrList<CPtrList, CEntity*> m_BackEnSet;
CPandaSrvrItem* GetEmbeddedItem() return (CPandaSrvrItem*)COleServerDoc::GetEmbeddedIte

virtual BOOL OnNewDocument();
virtual void Serialize(CArchive& ar);
virtual BOOL OnOpenDocument(LPCTSTR lpszPathName);
virtual BOOL OnSaveDocument(LPCTSTR lpszPathName);

protected:
virtual COleServerItem* OnGetEmbeddedItem();

public:
void UpdateBackgroundBoundary(CPoint);
void UpdateMapMax();
void UpdateMapMax(CEntity*);
void OpenTemplate();
void ClearBackgroundEntities();
int GetSubsetInFunction(CString, CArray<int,int>&);
void UpdateInfrustructure(CArray<CEntity*,CEntity*>&);
BOOL OpenAttachedFiles(LPCTSTR);
BOOL SaveAttachedFiles(LPCTSTR);
BOOL ReadDatabase();
BOOL CheckDatabase(const CString&, const CString&, const CStringArray&);
BOOL SaveDatabase();
BOOL OpenDatabase(CString, CString, CString, CStringArray&, int&);
BOOL OpenArcView(LPCTSTR);
void DoScale(double,double,double);
int CutEntity(int, CPoint);
BOOL ShowEntityXYData(int);
```

```
    BOOL StickEntity(int, BOOL, int, BOOL);
    BOOL InsertEntity(CEntity*, int, BOOL);
    CEntity* DeleteEntity(int);
    void OrderEntity(int,int);
    double FindNearestDis(CRealPoint, int&, int&);
    CEntity* FindEntity(int);
    void ReInitializeData();
    virtual C̃PandaDoc();
    #ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
    #endif

    protected:
    virtual CDocObjectServer* GetDocObjectServer(LPOLEDOCUMENTSITE
pDocSite);
    afx_msg void OnProjectSettings();
    afx_msg void OnOperationprofileComposition();
    afx_msg void OnOperationprofileFugitivefueltanks();
    afx_msg void OnOperationprofileGseandservivefuel();
    afx_msg void OnOperationprofileInterroadtraffic();
    afx_msg void OnOperationprofileMotorwaytraffic();
    afx_msg void OnOperationprofileRunwayoperationtime();
    afx_msg void OnOperationprofileRunwayusage();
    afx_msg void OnProjectDatabase();
    afx_msg void OnProjectIndicatorlimits();
    afx_msg void OnOperationprofileNoise();
    afx_msg void OnNoiseAcesssamples();
    afx_msg void OnOperationprofileCarpark();
    afx_msg void OnOperationprofileSurpasdistribution();
    afx_msg void OnProjectInfrustructure();
    afx_msg void OnOperationprofileAirportroadtraffic();
    afx_msg void OnOperationprofileAirportemployeesandextrasurfacepassengers();
    afx_msg void OnProjectMonitoringdata();
    afx_msg void OnOperationsRatiomotorwaytraffic();
    afx_msg void OnOperationsRatiocarpartraffic();
    afx_msg void OnProjectUpdatetemplate();
    afx_msg void OnFileSaveAs();
    DECLARE_MESSAGE_MAP()
    DECLARE_DISPATCH_MAP()
    DECLARE_INTERFACE_MAP()

    private:
    int m_FileOption;
    };
```

C.2.2 CPandaView

```
class CPandaView : public CScrollView
    {
    protected:
    CPandaView();
    DECLARE_DYNCREATE(CPandaView)

    public:
    CPandaCntrItem* m_pSelection;
    CPandaDoc* GetDocument();
    virtual void OnDraw(CDC* pDC);
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);

    protected:
    virtual void OnInitialUpdate();
    virtual BOOL OnPreparePrinting(CPrintInfo* pInfo);
    virtual void OnBeginPrinting(CDC* pDC, CPrintInfo* pInfo);
    virtual void OnEndPrinting(CDC* pDC, CPrintInfo* pInfo);
    virtual void OnPrint(CDC* pDC, CPrintInfo* pInfo);
    virtual BOOL IsSelected(const CObject* pDocItem) const;
    virtual void OnActivateView(BOOL bActivate, CView* pActivateView, CView*
pDeactiveView);

    public:
    double m_ViewScale;
    BOOL m_BkMapLoaded;
    CDC m_BkMapDc;
    BOOL m_ShowBackEntity;
    CStringArray m_MonitorSet;
    int m_OldBackOption;
    double m_ScaleTheme;
    int m_BackgroundOption;
    CDC m_dcMem;
    int m_NonSpatialDrawAction;
    int m_DrawMode;
    int m_idRoadStage;
    CString m_Indicator;
    int m_Time;
    CZValueEdit* m_ZValueEdit;
    CPoint m_MovePoint1;
    double m_CurrentAngle;
    BOOL m_InRotate;
    RECT m_CurrentRect;
    BOOL m_OnStick;
    CPoint m_OldPoint;
    CEntity* m_TempEntitySet;
    BOOL m_BeginMove;
    BOOL m_RBStatus;
    BOOL m_LBStatus;
    CSelIndicator m_SelObject;
    CPoint m_MovePoint;
    UINT m_KeyboardStatus;
    CArray<int,int> m_Selected;
    CEntity m_CurrentEntity;
    BOOL m_NewEntity;
    CPoint m_CurrentPoint;
    void GetGridStructure(int&,int&,int&,int&,int&,int&);
    BOOL ReadINMNoiseContour(CString, CString);
    BOOL GetEmissionMonitorValue();
    BOOL GetNoiseMonitorValue(BOOL);
    BOOL DefineEntityProperty(CEntity*);
    double GetFuzzyMembership(double, double, double);
    BOOL DefineGreyNodes(CEntity*);
    void GetIntervalThrust(CString interval, double& min, double& max);
    void GetMapGrid(CArray<CRealPoint, CRealPoint>&, CArray<double, double>&,
BOOL IsArea=FALSE);
    BOOL GetDispersion(CArray<CRealPoint, CRealPoint>&, CArray<double,double>&,
int, int, CString&,
    BOOL IsPeak=FALSE);
    CSize DrawThemeMap(CDC*);
    void PrepareThemeMap(CArray<double,double>&, BOOL);
    int GetContourNode(int i, int j, int, double, double, double, CArray<double,
double>&, CArray<CRealPoint,CRealPoint>&);
```

```
void GetGridNode(int, int, double, double, double*, double*);
CEntity GetContour(CDC*, double, CArray<double,double>&, int, COLORREF, BOOL);
void DrawSelMark(CDC*, CPoint&);
void DrawSelMark(CDC*, int, int);
void DrawColourMark(int);
BOOL RotateWithSel(CDC*, CPoint);
void DoScale(double,double,double);
void InvalidatePart();
void RemoveSelMark();
BOOL DoStickEntity(CPoint);
RECT FindGroupMaxRect(int);
void MoveSize(CDC*, CPoint&);
void MoveWithSel(CDC*, CPoint&);
void MoveWithCreate(CDC*, CPoint&);
BOOL CutPath(CPoint&);
BOOL InsertNode(CPoint&);
int CheckConnection(CRealPoint&);
void ChangeSelOrder();
void AdjustRectCordinates(int,int,int,int,int&,int&,int&,int&);
void SelectEntity(CPoint);
void CreateEntity(CPoint);
void StopForAction();
virtual  CPandaView();
#ifdef _DEBUG
virtual void AssertValid() const;
virtual void Dump(CDumpContext& dc) const;
#endif

protected:
afx_msg void OnDestroy();
afx_msg void OnSetFocus(CWnd* pOldWnd);
afx_msg void OnSize(UINT nType, int cx, int cy);
afx_msg void OnInsertObject();
afx_msg void OnCancelEditCntr();
afx_msg void OnCancelEditSrvr();
afx_msg void OnLButtonDown(UINT nFlags, CPoint point);
afx_msg void OnLButtonDblClk(UINT nFlags, CPoint point);
afx_msg void OnMouseMove(UINT nFlags, CPoint point);
afx_msg void OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags);
afx_msg void OnKeyUp(UINT nChar, UINT nRepCnt, UINT nFlags);
afx_msg void OnLButtonUp(UINT nFlags, CPoint point);
afx_msg void OnContextMenu(CWnd* pWnd, CPoint point);
afx_msg void OnMapEdit();
afx_msg void OnMapEditArc();
afx_msg void OnMapEditBackward();
afx_msg void OnMapEditCurve();
afx_msg void OnMapEditCurveclosed();
afx_msg void OnMapEditCurvepath();
afx_msg void OnMapEditCut();
afx_msg void OnMapEditEllipse();
afx_msg void OnMapEditForward();
afx_msg void OnMapEditLine();
afx_msg void OnMapEditPath();
afx_msg void OnMapEditPoint();
afx_msg void OnMapEditPolygon();
afx_msg void OnMapEditPopClose();
afx_msg void OnMapEditPopColour();
afx_msg void OnMapEditPopStyle();
afx_msg void OnMapEditPopZedit();
afx_msg void OnMapEditRectangle();
afx_msg void OnMapEditReleaseRing();
afx_msg void OnMapEditRing();
afx_msg void OnMapEditRotate();
afx_msg void OnMapEditSelect();
afx_msg void OnMapEditStick();
afx_msg void OnMapEnlarge();
afx_msg void OnMapShrink();
afx_msg void OnUpdateMapEditRing(CCmdUI* pCmdUI);
afx_msg void OnUpdateMapEditStick(CCmdUI* pCmdUI);
afx_msg void OnUpdateMapEditReleaseRing(CCmdUI* pCmdUI);
afx_msg void OnUpdateMapEditRotate(CCmdUI* pCmdUI);
afx_msg void OnUpdateMapEditCut(CCmdUI* pCmdUI);
afx_msg void OnUpdateMapEditPopZedit(CCmdUI* pCmdUI);
afx_msg void OnUpdateMapEdit(CCmdUI* pCmdUI);
afx_msg void OnUpdateMapEnlarge(CCmdUI* pCmdUI);
afx_msg void OnUpdateMapShrink(CCmdUI* pCmdUI);
afx_msg void OnVIEWEntity();
afx_msg void OnViewContour();
afx_msg void OnMapEditPopDatabase();
afx_msg void OnMapEditPopLineColour();
afx_msg void OnMapEditPopLineStyle();
afx_msg void OnMapEditPopLineWidth();
afx_msg void OnUpdateMapEditPopColour(CCmdUI* pCmdUI);
afx_msg void OnUpdateMapEditPopClose(CCmdUI* pCmdUI);
afx_msg void OnUpdateMapEditPopStyle(CCmdUI* pCmdUI);
afx_msg void OnMapEditPopBackgroundcolour();
afx_msg void OnUpdateMapEditPopBackgroundcolour(CCmdUI* pCmdUI);
afx_msg void OnEmissionsAircraftassociatedemission();
afx_msg void OnEmissionsAirportservice();
afx_msg void OnEmissionsCarparkcoldstart();
afx_msg void OnEmissionsGroundsupportequipment();
afx_msg void OnEmissionsInterroademission();
afx_msg void OnEmissionsMotorwayemission();
afx_msg void OnEmissionsTankfugitiveemission();
afx_msg void OnViewActiveemission();
afx_msg void OnViewAircraftemission();
afx_msg void OnViewAirportservice();
afx_msg void OnViewCarparkcoldstartemission();
afx_msg void OnViewFugitiveemission();
afx_msg void OnViewGroundsupportequipment();
afx_msg void OnViewInterroademission();
afx_msg void OnViewMotorwayemission();
afx_msg void OnAiporttoolTerminal();
afx_msg void OnAirporttoolCarpark();
afx_msg void OnAirporttoolMonitor();
afx_msg void OnAirporttoolResident();
afx_msg void OnAirporttoolRoad();
afx_msg void OnAirporttoolRotate();
afx_msg void OnAirporttoolRunway();
afx_msg void OnAirporttoolSensitivearea();
afx_msg void OnAirporttoolTank();
afx_msg void OnAirporttoolTrajactory();
afx_msg void OnAirporttoolMotorway();
afx_msg void OnViewMonitoringNoiseCurve();
afx_msg void OnNoiseDistribution();
afx_msg void OnViewClearBackground();
afx_msg void OnViewShowBackground();
afx_msg void OnEmissionsDispersion();
afx_msg void OnAirporttoolGse();
afx_msg void OnEmissionsCarpark();
afx_msg void OnViewCarparkemission();
afx_msg void OnAirporttoolTaxiway();
afx_msg void OnUpdateEmissionsDispersion(CCmdUI* pCmdUI);
afx_msg void OnUpdateNoiseDistribution(CCmdUI* pCmdUI);
afx_msg void OnOtherindicatorsAircraftfuel();
afx_msg void OnViewAircraftfuelconsumption();
afx_msg void OnOtherindicatorsAirportroadfuel();
afx_msg void OnOtherindicatorsMotorwayfuel();
afx_msg void OnViewRoadfuelconsumption();
afx_msg void OnViewMotorwayfuelconsumption();
afx_msg void OnOtherindicatorsWaste();
afx_msg void OnViewWaste();
afx_msg void OnSustainabilityindicatorsRelativeindicators();
afx_msg void OnViewRelativeindicators();
afx_msg void OnNoiseNoiselevel();
afx_msg void OnNoiseNoisefrequency();
afx_msg void OnViewNoisefrequency();
afx_msg void OnViewNoiselevel();
afx_msg void OnProjectSpatial();
afx_msg void OnOperationsCheckcapacity();
afx_msg void OnSpatialEmissiondispersionindicators();
afx_msg void OnSpatialNoisefrequencyindicators();
afx_msg void OnSpatialNoiselevelindicators();
afx_msg void OnViewMonitordispersion();
afx_msg void OnViewMonitornoisefrequency();
afx_msg void OnViewMonitornoiselevel();
afx_msg void OnUpdateViewShowBackground(CCmdUI* pCmdUI);
afx_msg void OnUpdateViewClearBackground(CCmdUI* pCmdUI);
afx_msg void OnUpdateProjectSpatial(CCmdUI* pCmdUI);
afx_msg void OnUpdateSpatialEmissiondispersionindicators(CCmdUI* pCmdUI);
afx_msg void OnUpdateSpatialNoisefrequencyindicators(CCmdUI* pCmdUI);
afx_msg void OnUpdateSpatialNoiselevelindicators(CCmdUI* pCmdUI);
afx_msg void OnToolsShowbackentitys();
afx_msg void OnToolsClearbackentitys();
afx_msg void OnUpdateToolsClearbackentitys(CCmdUI* pCmdUI);
afx_msg void OnUpdateToolsShowbackentitys(CCmdUI* pCmdUI);
afx_msg void OnToolsMergebackentities();
afx_msg void OnUpdateToolsMergebackentities(CCmdUI* pCmdUI);
afx_msg void OnFileOpenexternalfiles();
afx_msg void OnUpdateFileOpenexternalfiles(CCmdUI* pCmdUI);
afx_msg void OnIndicatorsEdmsemission();
afx_msg void OnViewEdmsemissions();
afx_msg void OnSpatialInmcontours();
afx_msg void OnSpatialInmindicator();
afx_msg void OnViewInmnoiselevel();
afx_msg void OnUpdateSpatialInmcontours(CCmdUI* pCmdUI);
afx_msg void OnUpdateSpatialInmindicator(CCmdUI* pCmdUI);
afx_msg void OnViewBackgroundmap();
afx_msg void OnUpdateViewBackgroundmap(CCmdUI* pCmdUI);
afx_msg void OnBackgroundBoundary();
afx_msg void OnMapEditPopUncertainty();
afx_msg void OnMapEditPopThrust();
afx_msg void OnUpdateMapEditPopThrust(CCmdUI* pCmdUI);
afx_msg void OnMapEditPopGreynodes();
afx_msg void OnNoiseSpatialLevelSustainability();
DECLARE_MESSAGE_MAP()
private:
CString m_ObjectFunction;
};
```

Appendix D

Author's relevant publications

Some contents discussed in this thesis have been published in academic conferences or journals.

- Neural networks and its application: [205, 210, 209, 220, 219, 208, 218, 217, 229, 227, 228, 226, 222, 224, 225, 223]
- Fuzzy sets: [211, 212, 213]
- Rough sets: [215, 216, 213, 212]
- Grey sets: [202, 213, 212, 211, 203]
- GIS: [207, 205, 220, 218]
- Decision support: [206, 205, 187, 180, 221, 204]

Bibliography

[1] Arcview. URL = http://www.esri.com/software/arcview/, 2006.

[2] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions with Formulas Graphs and Mathematical Tables.* Dover, New York, USA, 1972.

[3] C. Adami. Airport real-time gis system. In *1997 ESRI User Conference*, San Diego, California, (USA), 1997.

[4] Federal Aviation Administration. *INM 6.0 Technical Manual.* FAA, Washington, DC, 2002.

[5] O. Ahlqvist, J. Keukelaar, and K. Oukbir. Rough classification and accuracy assessment. *International Journal of Geographical Information Science*, 14:475–496, 2000.

[6] O. Ahlqvist, J. Keukelaar, and K. Oukbir. Rough and fuzzy geographical data integration. *International Journal of Geographical Information Science*, 17:223–234, 2003.

[7] J. M. Airey, J. H. Rohlf, and F. P. Brooks. Towards image realism with interactive update rates in complex virtual building environments. *ACM Siggraph*, 24:41–50, 1990.

[8] B. Allenby. Industrial ecology, information and sustainability. *Foresight*, 2(2):163–171, 2000.

[9] C. Anderson, S. Augustine, Embt, and T. Thrasher. *Emission and dispersion modelling systems (EDMS) reference manual.* U.S. Dept. of Trans., Federal Aviation Administration, Report No. FAA-AEE-97-01, Washington, D.C, USA, 1997.

[10] R. Andrews, J. Diederich, and A. B. Tickle. A survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems*, 8(6):373–389, 1995.

[11] R. Andrews and S. Geva. Rule extraction from a constrained error back propagation mlp. In *Proc. 5th Australian Conference on Neural Networks*, volume 1, pages 9–12, Brisbane, Queensland, 1994.

[12] Anon. Indianapolis airport simplifies its payroll by computer. *American City and County*, 98:31–32, 1983.

[13] K.T. Antanassov. Intuitionistic fuzzy sets. In *VII ITKR's session*, Sofia, June 1983. deposed in Central Sci.-Technical Library of Bulg. Acad. of Sci., 1967/84.

[14] N. Ashford and P.H. Wright. *Airport engineering*. John Wiley & Sons, 1992.

[15] K. T. Atanassov. Intuitionistic fuzzy sets. *Fuzzy sets and systems*, 20:87–96, 1986.

[16] K. T. Atanassov. More on intuitionistic fuzzy sets. *Fuzzy sets and systems*, 33:37–46, 1989.

[17] K.T. Atanassov. *Intuitionistic fuzzy sets*. Physica-Verlag, Heidelberg - New York, 1999.

[18] N. Baba. A hybrid algorithm for finding the global minimum of error function of neural networks and its applications. *Neural Networks*, 7(8):1253–1265, 1994.

[19] M. Banerjee and S. K. Pal. Roughness of a fuzzy set. *Information Sciences*, 93:235–246, 1996.

[20] A.D. Basiago. Sustainable development in british land use regulation. *World Resource Review*, 7:570–588, 1995.

[21] D. P. Bertsekas and J. N. Tsitsiklis. *Introduction to Probability*. Athena Scientic, Belmont - Massachusetts, 2002.

[22] W. Blum. Sustainable landuse and soil and water conservation: A perspective for 50 more years. In *Proceedings of the 50th annual meeting of the soil and water conservation society*, page 8, Ankeny, 1995.

[23] J.C. Bobick. Evolution and future of simmod. In *Proceedings of The Airport Modelling and Simulation Conference*, pages 233–239, Arlington, Virginia, (USA), 1999.

[24] S. Bodjanova. Alpha-bounds of fuzzy numbers. *Information Sciences*, 152:237–266, 2003.

[25] A. Bosshard. A methodology and terminology of sustainability assessment and its perspectives for rural planning. *Agriculture, Ecosystems & Environment*, 77:29–41, 2000.

[26] C. Brunsdon and S. Openshaw. Error simulation in vector gis using neural computing methods. In M. F. Worboys, editor, *Innovation in GIS*, pages 177–200. Taylor & Francis Ltd, London, 1994.

[27] P. Burillo and H. Bustince. Entropy on intuitionistic fuzzy sets and on interval-valued fuzzy sets. *Fuzzy Sets and Systems*, 78:305–316, 1996.

[28] P. A. Burrough. Gis and geostatistics: essential partners for spatial analysis. In *Proceedings of the international symposium on spatial data quality'99*, pages 10–20, 1999.

[29] P.A. Burrough and R.A. McDonnell. *Principles of Geographical Information Systems.* Oxford University Press, Oxford, 1998.

[30] H. Bustince and P. Burillo. Vague sets are intuitionistic fuzzy sets. *Fuzzy Sets and Systems*, 79:403–405, 1996.

[31] E.J. Calderon. An applied method for the assessment of sustainability of urban pilot projects. *Environmental Impact Assessment Review*, 20(3):289–298, 2000.

[32] G. A. Carpenter and S. Grossberg. Adaptive resonance theory. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 87–90. MIT Press, Cambridge, MA, 2003.

[33] D. Carruthers, R. Holroyd, J. Hunt, W. Weng, A. Robins, D. Apsley, D. Thomson, and F. Smith. Uk-adms: A new approach to modelling dispersion in the earth's atmospheric boundary layer. *Journal of Wind Engineering and Industrial Aerodynamics*, 52:139–153, 1994.

[34] NASA Langley Research Center. Aircraft noise prediction program. URL = http://www.sae.org/technical/papers/891032, 2006.

[35] R. Chapkis. Impact of technical differences between the methods of integrated noise model and noisemap. In *Proceedings of the "INTER-NOISE 80"*, volume 2, pages 831–834, Miami, FL, (USA), 1980.

[36] B. Chazelle. *Application Challenges to Computational Geometry.* Technical Report TR-521-96, Princeton University, 1996.

[37] F. J. Cheng, S. H. Hui, and Y. C. Chen. Reservoir operation using grey fuzzy stochastic dynamic programming. *Hydrological Processes*, 16:2395–2408, 2002.

[38] Y. Cheng. Rule-based reactive model for the simulation of aircraft on airport gates. *Knowledge-Based Systems*, 10:225–236, 1998.

[39] J. Y. Chiaoa, A. R. Bordeauxa, and N. Ambady. Mental representations of social status. *Cognition*, 93:B49–B57, 2004.

[40] N. R. Chrisman. The error component in spatial data. In D. J. Maguire, M. F. Goodchild, and D. Rhind, editors, *Geographical information systems: principles and applications.* Longman Scientific & Technical, John Wiley, New York (USA), 1991.

[41] T. Connor. Integrated noise model - the federal aviation administration's computer program for predicting noise exposure around an airport. In *Proceedings of the "INTER-NOISE 80"*, volume 1, pages 127–130, Miami, FL, (USA), 1980.

[42] C. Cornelis, K. T. Atanassov, and E. E..Kerre. Intuitionistic fuzzy sets and interval-valued fuzzy sets: a critical comparison. In M. Wagenknecht and R. Hampel, editors, *Third EUSFLAT proceedings*, pages 159–163, Zittau, Germany, September 2003. European Society for Fuzzy Logic and Technology.

[43] Caliper Corporation. Transcad. URL = http://www.caliper.com/tcovu.htm, 2006.

[44] V. Cross and A. Firat. Fuzzy objects for geographical information systems. *Fuzzy Sets and Systems*, 113:19–36, 2000.

[45] R. Deloach. Airport noise impact reduction through operations. In *Presented at Aircraft Safety and Operating Problems Conf., Hampton, Va., 5-7 Nov. 1980*, pages 5–7, 1981.

[46] S. Demirkale. The measurement and analysis of the noise caused by run-up operations at atatuerk airport. *Architectural Science Review*, 39:39–47, 1996.

[47] J. Deng. The control problems of grey systems. *Systems and Control Letters*, 1982.

[48] G. Deschrijver, C. Cornelis, and E. E. Kerre. On the representation of intuitionistic fuzzy t-norms and t-conorms. *IEEE Transactions on Fuzzy Ssystems*, 12(1):45–61, 2004.

[49] G. Deschrijver and E.E. Kerre. On the relationship between some extensions of fuzzy set theory. *Fuzzy sets and systems*, 133(2):227–235, 2003.

[50] D. Dubois, W. Ostasiewicz, and H. Prade. Fuzzy set: history and basic notions. In D. Dubois and H. Prade, editors, *Fundamentals of fuzzy sets*, pages 21–124. Kluwer, 2000.

[51] D. Dubois and H. Prade. Rough sets and fuzzy rough sets. *International journal of general systems*, 17(191–209), 1990.

[52] D. Dubois and H. Prade. Fuzzy relation equations and causal reasoning. *~~Fuzzy sets and~~ systems*, 75:119–134, 1995.

[53] D. Dubois, H. Prade, F. Esteva, P. Garcia, and L. Godo. A logical approach to interpolation based on similarity relations. *International Journal of Approximate Reasoning*, 17:1–36, 1997.

[54] W. Duch, R. Adamczak, and K. Grabczewski. Extraction of logical rules from training data using backpropagation networks. In *Proc. of the First Online Workshop on Soft Computing*, pages 25–30, 1996.

[55] K.M. Eldred. Model for airport noise exposure on a national basis - 1960 to 2000. In *Proceedings of the "INTER-NOISE 80"*, volume 2, pages 803–808, Miami, FL, (USA), 1980.

[56] FAA. Emissions and dispersion modeling system. URL = http://www.faa.gov/about/office_org/headquarters_offices/aep/models/edms_model/, 2006.

[57] FAA. Heliport noise model. URL = http://www.faa.gov/about/office_org/headquarters_offices/aep/models/hnm_model/, 2006.

[58] P.F. Fisher. First experiments in viewshed uncertainty: simulating the fuzzy viewshed. *Photogrammetric Engineering and Remote Sensing*, 58:345–352, 1992.

[59] P.F. Fisher. Algorithm and implementation uncertainty in the viewshed function. *International Journal of Geographical Information Systems*, 7:331–347, 1993.

[60] P.F. Fisher. Probable and fuzzy models of the viewshed operation. In M. F. Worboys, editor, *Innovation in GIS*, pages 161–175. Taylor & Francis Ltd, London, 1994.

[61] P.F. Fisher. An exploration of probable viewsheds in landscape planning. *Environment and Planning B: Planning and Design*, 22:527–546, 1995.

[62] G.P. Fletcher and C.J. Hinde. Using neural networks as a tool for constructing rule based systems. *Knowledge Based Systems*, 8(4):183–189, 1995.

[63] G.P. Fletcher and C.J. Hinde. Producing evidence for the hypoitheses of large neural networks. *Neurocomputing*, 10:359–373, 1996.

[64] C. C. Fonte and W. Lodwick. Areas of geographical entities. *International Journal of Geographical Information Science*, 18:127–150, 2004.

[65] A. Frail. Airport noise modelling and aircraft scheduling so as to minimize community annoyance. *Applied Mathematical Modelling*, 8(4):271–281, 1984.

[66] L. Frair. Airport noise modelling and aircraft scheduling so as to minimize community annoyance. *Applied Mathematical Modelling*, 8:271–281, 1984.

[67] T. Funkhouser and C. Squin. Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments. *Computer Graphics*, 27:247–254, 1993.

[68] M.R. Gatersleben, W. Van, and W. Simon. Analysis and simulation of passenger flows in an airport terminal. In *Winter Simulation Conference Proceedings*, volume 2, pages 1226–1231, Phoenix, AZ, (USA), 1999.

[69] GIS/Trans. Gia-t. URL = http://www.gistrans.com, 2006.

[70] C. M. Gold. Problems with handling spatial data-the voronoi approach. *CISM Journal*, 45:65–80, 1991.

[71] D. Goldberg. *Genetic algorithms in search, optimisation and machine learning.* Addison-Welsley, Reading, MA, 1989.

[72] M. F. Goodchild. Geographic information systems and cartography. *Cartography*, 19:1–13, 1990.

[73] M. F. Goodchild. Issues of quality and uncertainty. In J. C. Muller, editor, *Advances in Cartography*, pages 113–139. Elsevier Applied Science, London, England, 1991.

[74] M.F. Goodchild. Fractals and the accuracy of geographical measures. *Mathematical Geology*, 12:85–98, 1980.

[75] M.F. Goodchild. Measurement-based gis. In *Proceedings of the international symposium on spatial data quality*, pages 1–9, 1999.

[76] M.F. Goodchild and S. Gopal. *The accuracy of spatial databases.* Taylor & Francis, London, 1989.

[77] M.F. Goodchild, G. Sun, and S. Yang. Development and test of an error model for categorical data. *International Journal of Geographical Information Systems*, 6:87–104, 1992.

[78] B. Graham and C. Guyer. Environmental sustainability, airport capacity and european air transport liberalization: irreconcilable goals? *Journal of Transport Geography*, 7:165–180, 1999.

[79] P. Grzegorzewski. Distances between intuitionistic fuzzy sets and/or interval-valued fuzzy sets based on the hausdorff metric. *Fuzzy Sets and Systems*, 148:319–328, 2004.

[80] H. W. Guesgen and J. Albrecht. Imprecise reasoning in geographic information systems. *Fuzzy Sets and Systems*, 113:121–131, 2000.

[81] J. Habermas. *Communication and the Evolution of Society.* Polity Press, London, 1979.

[82] A. Hagen. Fuzzy set approach to assessing similarity of categorical maps. *International Journal of Geographical Information Science*, 17:235–249, 2003.

[83] M. Hagiwara. Removal of hidden units and weights for back propagation networks. In *Proceedings of IJCNN*, volume 1, pages 351–354, Nagoya, Japan, 1993.

[84] D. Hand, H. Mannila, and P. Smyth. *Principles of data mining.* MIT Press, Cambridge, MA, 2001.

[85] B. Hassibi and D. Stork. Second order derivatives for pruning: optimal brain surgeon. *Advances in Neural Information Processing Systems*, 5:164-171, 1993.

[86] F. Hausdorff. *Set theory.* Chelsea Publishing Company, New York, USA, 1962. Transl. by J. R. Aumann from the original German 'Mengenlehre'.

[87] S. Haykin. *networks: a comprehensive foundation.* Prentice-Hall, New Jersey, 1999.

[88] R. Hecht-Nielsen. Kolmogorov's mapping neural network existence theorem. In *Proc. IEEE 1st Int. Conf. on Neural Networks*, volume 3, pages 11–14, San Diego, 1987.

[89] R. Hecht-Nielsen. Theory of the back-propagation neural network. In *Proceedings of the International Joint Conference on Neural Networks*, volume 1, pages 593–611, 1989.

[90] R. Hecht-Nielsen. *Neurocomputing.* Addison-Welsley, Reading, MA, 1990.

[91] G.B. Heuvelink and P. A. Burrough. Error propagation in cartographic modelling using boolean logic and continuous classification. *International Journal of Geographical Information Systems*, 7:231–246, 1993.

[92] J. J. Hopfield. Neurons with graded response have collective computational properties like those of two state neurons. In *Proceedings of the US National Academy of Science*, volume 81, pages 3088–3092, 1984.

[93] G. A. Horridge. Optical mechanisms of clear-zone eyes. In *The compound eye and vision of insects*, pages 255–298. Clarendon Press, Oxford, 1975.

[94] W.L. Hung and M.S. Yang. Similarity measures of intuitionistic fuzzy sets based on hausdorff distance. *Pattern Recognition Letters*, 25:1603–1611, 2004.

[95] D. W. Hutchison and S. D. Hill. Simulation optimization of airline delay using simultaneous perturbation stochastic approximation. In *Proceedings of IEEE Annual Simulation Symposium (SS 2000)*, pages 253–258, Washington, DC, (USA), 2000.

[96] T. B. Iwinski. Algebraic approach to rough sets. *Bulletin of the polish Academy of Science and mathematics*, 35:673–683, 1987.

[97] R. Jensen and Q. Shen. Fuzzy rough attribute reduction with application to web categorization. *Fuzzy sets and systems*, 141:469–485, 2004.

[98] T. Jones. Globalization and environmental sustainability: an oecd perspective. *International Journal of Sustainable Development*, 3(2):146–158, 2000.

[99] J. Kacprzyk. *Multistage fuzzy control.* Wiley, Chichester, 1997.

[100] J. N. Kapur and H. K. Kesavan. Jaynes' maximum entropy principle. In *Entropy optimisation principles with applications*, pages 23–76. Academic Press, San Diego, 1992.

[101] J. M. Karpilow, A. C. Pimentel, H. K. Shamloula, and T. R. Venkatesh. Neuronal development in the drosophila compound eye: Photoreceptor cells r1, r6, and r7 fail to differentiate in the retina aberrant in pattern (rap) mutant. *Journal of Neurobiology*, 31(2):149–165, 1996.

[102] N. Kasabov. Learning fuzzy rules and approximate reasoning in fuzzy neural networks and hybrid systems. *Fuzzy Sets and Systems*, 82:2–20, 1996.

[103] R. L. Keeney and H. Raiffa. *Decisions with multiple objectives: preference and value tradeoffs.* John Wiley & Sons, New York, 1976.

[104] G. J. Klir and B. Yuan. *Fuzzy sets and fuzzy logic theory and applications.* Prentice Hall P T R, Upper Saddle River - New Jersey, 2002.

[105] P. Koiren and E. D. Sontag. Neural networks with quadratic vc dimension. *Advances in Neural Information Processing Systems*, 8:197–203, 1996.

[106] J. Komorowski, J. Z. Pawlak, L. Polkowski, and A. Skowron. Rough sets: a tutorial. In S. K. Pal and A. Skowron, editors, *Rough fuzzy hybridization: a new trend in decision-making.* Springer, Singapore, 1999.

[107] CM. Kootwijk-Damman and PA. Van Wijngaarden. Noise disturbance from railway traffic best tackled at source. *Rail International*, 28:35–40, 1997.

[108] Wyle Laboratories. Noise model simulation. URL = http://www.wylelabs.com/products/acousticsoftwareproducts/nmsim.html, 2006.

[109] R. Laurini and D. Thompson. *Fundamentals of spatial information systems.* Academic Press, London, 1992.

[110] Y. Leung. On the imprecision of boundaries. *Geographical Analysis*, 19:125–151, 1987.

[111] Y. Leung. *Intelligent spatial decision support systems.* Springer, Berlin/Heidelberg, 1997.

[112] Deng-Feng Li. Multattribute decision making models and methods using intuitionistic fuzzy sets. *Journal of Computer and Systems Sciences*, 70:73–85, 2005.

[113] S. Liu, T. Gao, and Y. Dang. *Grey systems theory and its applications.* The Science Press of China, Beijing, 2000.

[114] W.A. Lodwick, W. Monson, and L. Svoboda. Attribute error and sensitivity analysis of map operations in geographical information systems. *International Journal of Geographical Information Systems*, 4:413–427, 1990.

[115] A. Lozowski, T. J. Cholewo, and J. M. Zurada. Crisp rule extraction from perceptron network classifiers. In *Proc. of the 1996 IEEE International Conference on Neural Networks Plenary, Panel and Special Sessions*, volume 1, pages 94–99, Washington DC, 1996.

[116] N. Majima, A. Watanabe, A. Yoshimura, and T. Nagano. A new criterion "effectiveness factor" for pruning hidden units. In *Proceedings of IJCNN*, volume 1, pages 382–385, Seoul, Korea, 1994.

[117] J. Malczewski. *GIS and multicriteria decision analysis.* John Wiley & Sons, New York, 1999.

[118] J. Mao, K. Mohiuddin, and A. Jain. Parsimonious network design and feature selection through node pruning. In *Proceedings of 12th International Conference on Pattern Recognition*, pages 622–624, 1994.

[119] R. Mato and T. Mufuruki. Noise pollution associated with the operation of the dar es salaam international airport. *Transportation Research, Part D: Transport and Environment*, 4D:81–89, 1999.

[120] R. Matsuhashi, K. Ishitani, P. Riemer, and A. Smith. Model analyses for sustainable energy supply taking resource and environmental constraints into consideration. *Energy Conversion & Management*, 37:1253–1258, 1995.

[121] B. Mcgrath. The sustainability of a car dependent settlement pattern: an evaluation of new rural settlement in ireland. *Environmentalist*, 19(2):99–107, 1999.

[122] McNerney and T. Michael. Airport pavement management as part of an integrated airport geographical information system (gis). In *Proceedings for The Transportation Congress*, volume 1, pages 573–587, New York, NY, (USA), 1995. ASCE.

[123] J. Mendel and R.I. John. Type-2 fuzzy sets made simple. *IEEE Transactions on Fuzzy Systems*, 10(2):117–127, 2002.

[124] METRON. Noise integrated routing system. URL = http://www.metronaviation.com/nirs.php, 2006.

[125] H.J. Miller and S.L. Shaw. *Geographic information systems for transportation (GIS-T): principles and applications.* Oxford University Press, Oxford, 2001.

[126] M. Molenaar and T. Cheng. Fuzzy spatial objects and their dynamics. *ISPRS Journal of Photogrammetry and Remote Sensing*, 55:164–175, 2000.

[127] P. Morrell and C. H. Lu. Aircraft noise social cost and charge mechanisms - a case study of amsterdam airport schiphol. *Transportation Research, Part D: Transport and Environment*, 5:305–320, 2000.

[128] A. Mousavi and P. Jabedar-Maralani. Double-faced rough sets and rough communication. *Information Science*, 148:41–53, 2002.

[129] M. C. Mozer and P. Smolensky. Skeletonization: a technique for trimming the fat from a network via relevance assessment. In D. S. Touretzky, editor, *Advances in Neural Information Processing*, pages 107–115. Denver, 1988.

[130] S. Nanda and S. Majumdar. Fuzzy rough sets. *Fuzzy sets and systems*, 45:157–160, 1992.

[131] A. Obtulowicz. Rough sets and heyting algebra valued sets. *Bull. Polish Acad. Sci. Math.*, 35:667–671, 1987.

[132] A. R. Odoni, J. Bowman, D. Delahaye, J. J. Deyst, E. Feron, R. J. Hansman, K. Khan, J. K. Kuchar, N. Pujet, and R.W. Simpson. Existing and required modeling capabilities for evaluating atm systems and concepts. In *Proceedings of The International Conference on Water Quality Modelling in the Inland Natural Environment*, http://web.mit.edu/aeroastro/www/labs/AATT/aatt.html, 1997.

[133] R.G. O'lone. Mass of dilemmas forcing operators to mesh airport use with environment. *Aviation Week and Space Technology*, 135:86–91, 1991.

[134] J. O'Rourke. *Computational geometry in C.* Cambridge University Press, 1994.

[135] S. K. Pal and A. Ghosh. Fuzzy geometry in image analysis. *Fuzzy sets and systems*, 48:23–40, 1992.

[136] V. Palade, D.C. Neagu, and R.J. Patton. Interpretation of trained neural networks by rule extraction. In B. Reusch, editor, *Fuzzy Days 2001, LNCS2206*, pages 152–161. Springer, 2001.

[137] J. Park and J. W. Sandberg. Universal approximation function using radial basis functions network. *Neural Computation*, 3:246–257, 1991.

[138] Z. Pawlak. Rough sets. *International journal of Computer and Information Sciences*, 11(5):341–356, 1982.

[139] Z. Pawlak. *Rough sets: theoretical aspects of reasoning about data.* Kluwer Academic Publishers, Dordrecht, 1991.

[140] Z. Pawlak. Rough sets and intelligent data analysis. *Information Science*, 147:1–12, 2002.

[141] F. E. Petry, M. A. Cobb, L. Wen, and H. Yang. Design of system for managing fuzzy relationships for integration of spatial data in querying. *Fuzzy Sets and Systems*, 140:51–73, 2003.

[142] J. R. Quinlan. Induction on decision trees. *Machine Learning*, 1:81–106, 1996.

[143] A.M. Radzikowska and E.E. Kerre. A comparative study of fuzzy rough sets. *Fuzzy sets and systems*, 126:137–155, 2002.

[144] D. Raper, J. Longhurst, and D. Conlan. The impact of airport operations on urban air quality. In C. A. BREBBIA, H. POWER, P. ZANNETTI, and J.M. BALDASANO, editors, *Air Pollution II Vol. 2: Pollution Control and Monitoring*, pages 91–98. Kluwer Academic, BILLERICA, MA, (USA), 1994.

[145] J. Ravetz. Integrated assessment for sustainability appraisal in cities and regions. *Environmental Impact Assessment Review*, 20(1):31–64, 2000.

[146] R. Reed. Pruning algorithms - a survey. *IEEE Transactions on Neural Networks*, 4(5):740–747, 1991.

[147] B.C. Richardson. Towards a policy on a sustainable transportation system. *Transportation Research Record*, 1670:27–34, 1999.

[148] J. P. Rigol, C. H. Jarvis, and N. Stuart. Artificial neural networks as a spatial interpolation. *International Journal of Geographical Information Science*, 15:323–343, 2001.

[149] S.E. Rodenbeck and M.L. Maslia. Groundwater modeling and gis to determine exposure to tce at tucson. *Practice Periodical of Hazardous, Toxic, and Radioactive Waste*, 2:53–61, 1998.

[150] A. Rosenfeld. Fuzzy digital topology. *Information and Control*, 40:76–87, 1979.

[151] D.E. Rumelhart, E.H. Geoffrey, and J.W. Ronald. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[152] J.M. Fernandez Salido and S. Murakami. Rough set analysis of a general type of fuzzy data using transitive aggregations of fuzzy similarity relations. *Fuzzy sets and systems*, 139:515–542, 2003.

[153] R. Sambuc. *Fonctions Φ -floues. Application l'Aide au Diagnostic en Pathololologie Thyroidienne.* PhD thesis, Univ. Marseille, France, 1975.

[154] G.M. Sandquist, D.M. Slaughter, and C.Y. Kimura. *Improved assessment of aviation hazards to ground facilities using a geographical information system.* Report: UCRL-JC-124357, CONF-960912-10, NTIS, 5285 Port Royal Rd, Springfield, VA 22161, USA, 1996.

[155] R. Setiono. A penalty function approach for pruning feed-forward neural networks. *Neural Computation*, 9:185–204, 1997.

[156] R. Setiono, W.K. Leow, and J.M. Zurada. Extraction of rules from artificial neural networks for nonlinear regression. *IEEE Transactions on Neural Networks*, 13(3):564-577, 2002.

[157] R. Setiono and H. Liu. Neurolinear: From neural networks to oblique decision rules. *Neurocomputing*, 17(1):1–24, 1997.

[158] R.R. Souleyrette and T.R. Strauss. Transportation. In S. Easa and Y. Chan, editors, *Planning & development applications of GIS*, pages 117–133. American Society of Civil Engineers, Reston, VA, (USA), 1999.

[159] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society*, B36:111–133, 1974.

[160] D.Z. Sui and M.F. Goodchild. Gis as media ? *International Journal of Geographical Information Science*, 15:387–390, 2001.

[161] E. Szmidt and J. Baldwin. New similarity measures for intuitionistic fuzzy set theory and mass assignment theory. *Notes on IFS*, 9(3):60–76, 2003.

[162] E. Szmidt and J. Baldwin. Entropy for intuitionistic fuzzy set theory and mass assignment theory. *Notes on IFS*, 10(3):15–286, 2004.

[163] E. Szmidt and J. Baldwin. Assigning the parameters for intuitionistic fuzzy sets. *Notes on IFS*, 11(6):1–12, 2005.

[164] E. Szmidt and J. Kacprzyk. Intuitionistic fuzzy sets in group decision making. *Notes on IFS*, 2:15–32, 1996.

[165] E. Szmidt and J. Kacprzyk. Remarks on some applications of intuitionistic fuzzy sets in decision making. *Notes on IFS*, 2(3):22–31, 1996.

[166] E. Szmidt and J. Kacprzyk. On measuring distances between intuitionistic fuzzy sets. *Notes IFS*, 3:1–13, 1997.

[167] E. Szmidt and J. Kacprzyk. Group decision making under intuitionistic fuzzy preference relations. In *IPMU'98 proceedings*, pages 172–178, Paris, La Sorbonne, 1998.

[168] E. Szmidt and J. Kacprzyk. Distance between intuitionistic fuzzy sets. *Fuzzy sets and systems*, 114(3):505-518, 2000.

[169] E. Szmidt and J. Kacprzyk. On measures of consensus under intuitionistic fuzzy relations. In *IPMU'2000 proceedings*, pages 641–647, Madrid, 2000.

[170] E. Szmidt and J. Kacprzyk. Analysis of consensus under intuitionistic fuzzy preferences. In *Proc. Int. Conf. in fuzzy logic and technology*, pages 79–82, De Montfort Univ. Leicester, UK, 2001.

[171] E. Szmidt and J. Kacprzyk. Entropy for intuitionistic fuzzy sets. *Fuzzy sets and systems*, 118(2):467–477, 2001.

[172] E. Szmidt and J. Kacprzyk. Analysis of agreement in a group of experts via distances between intuitionistic fuzzy preferences. In *IPMU'2002 proceedings*, pages 1859–1865, Annecy, France, 2002.

[173] E. Szmidt and J. Kacprzyk. Evaluation of agreement in a group of experts via distances between intuitionistic fuzzy sets. In *Proceedings of Int. IEEE Symposium: Intelligent Systems*, pages 166–170, Varna, Bulgaria, 2002.

[174] E. Szmidt and J. Kacprzyk. An intuitionistic fuzzy set based approach to intelligent data analysis (an application to medical diagnosis). In A. Abraham, L. Jain, and J. Kacprzyk, editors, *Recent advances in intelligent paradigms and applications*, pages 57–70. Springer-Verlag, 2002.

[175] E. Szmidt and J. Kacprzyk. A measure of similarity for intuitionistic fuzzy sets. In M. Wagenknecht and R. Hampel, editors, *Third EUSFLAT proceedings*, pages 206–209, Zittau, Germany, September 2003. European Society for Fuzzy Logic and Technology.

[176] E. Szmidt and J. Kacprzyk. Similarity of intuitionistic fuzzy sets and the jaccard coefficient. In *IPMU'2004 proceedings*, pages 1405–1412, Perugia, Italy, 2004.

[177] E. Szmidt and J. Kacprzyk. A new concept of a similarity measure for intuitionistic fuzzy sets and its use in group decision making. In V. Torra, Y. Narukawa, and S. Miyamoto, editors, *Modelling decisions for artificial intelligence. LNAI 3558*, pages 272–282. Springer, 2005.

[178] C. Tan and Q. Zhang. Fuzzy multiple attribute topsis decision making mrthod based on intuitionistic fuzzy set theory. In *IFSA '2005 proceedings*, pages 1602–1605, 2005.

[179] R. M. Tayler. A prediction model for airport ground noise propagation. In *proceedings of the fifteenth international conference on noise engineering (INTER-NOISE 86)*, volume 1, pages 431–436, 1986.

[180] C. Thomas, D. Raper, P. Upham, D. Gillingwater, Y. Yang, and C.J. Hinde. A strategic decision support tool for indicating airport sustainability. *Environmental Modelling and Software*, 16:297–298, 2001.

[181] A. B. Tickle, R. Andrews, M. Golea, and J. Diederich. The truth will come to light: diretctions and challenges in extracting the knowledge embedded within trained neural networks. *IEEE Transactions on Neural Networks*, 9(6):1057–1068, 1998.

[182] V. Tosic. A review of airport passenger terminal operations analysis and modelling. *TRANSP. RES.*, 26:3–26, 1992.

[183] G. Towell and J. Shavlik. The extraction of refined rules from knowledge based neural networks. *Machine Learning*, 131:71–101, 1993.

[184] L. Trana and L. Ducksteinb. Comparison of fuzzy numbers using a fuzzy distance measure. *Fuzzy Sets and Systems*, 130:331–341, 2002.

[185] M. Tylor, J.M. Denner, and P.W. Jowitt. De-icant pollution of airport run-off development of a simulation model for evaluation of run-off management strategies. In *Proceedings of The International Conference on Water Quality Modelling in the Inland Natural Environment*, pages 579–586, Bournemouth, 1986.

[186] R.M. Tylor. A prediction model for airport ground noise propagation. In *Proceedings of The Fifteenth International Conference on Noise Engineering (INTER-NOISE86)*, volume 1, pages 431–436, Cambridge, MA, (USA), 1986.

[187] P. Upham, Y. Yang, D. Raper, C. Thomas, D. Gillingwater, and C.J. Hinde. Mitigation environment constraints at airports through long term planning: a decision support approach. *Air Traffic Control Quarterly (An International Journal of Engineering and Operations)*, 12(2):107–124, 2004.

[188] M. Vallet. Annoyance after change in airport noise environment. In *Proceedings for INTER-NOISE 96*, page 95, Liverpool, 1996.

[189] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theoretical Probability and Its Applications*, 17:264-280, 1971.

[190] G. Viglioccoa, D. P. Vinsona, M. F. Damianb, and W. Levelt. Semantic distance effects on object and action naming. *Cognition*, 85:B61–B69, 2002.

[191] H. B. Voelcker, A. A. G. Requicha, and R. W. Conway. Computer applications in manufacturing. In J. F. Traub, editor, *Annual Review of Computer Science*, pages 349–387. Annual Reviews Inc., Palo Alto, CA, 1988.

[192] H. B. Voelcker, A. A. G. Requicha, and R. W. Conway. Research in solid modeling at the university of rochester: 1972-1987. In L. Piegl, editor, *Fundamental Developments of Computer-Aided Geometric Modeling*, pages 203–254. Academic Press Ltd, London, England, 1993.

[193] F. Wang, G. B. Hall, and Subaryono. Fuzzy information representation and processing in conventional gis software: database design and application. *International Journal of Geographical Information Systems*, 4:261–283, 1990.

[194] G.J. Wang and Y.H. He. Intuitionistic fuzzy sets and l-fuzzy sets. *Fuzzy Sets and Systems*, 110:271–274, 2000.

[195] I. D. Wilson, J. M. Ware, and J. A. Ware. A genetic algorithm approach to cartographic map generalisation. *Computers in Industry*, 52:291–304, 2003.

[196] A. Wiweger. On topological rough sets. *Bull. Polish Acad. Sci. Math.*, 37:89-93, 1989.

[197] T. Wolff and D. F. Ready. Pattern formation in the drosophila retina. In *The Development of Drosophila melanogaster*, volume 2, pages 1277–1325. Cold Spring Harbor Laboratory Press, 1993.

[198] M.F. Worboys, H.M. Hearnshaw, and D.J. Maguire. Object-oriented data modelling for spatial databases. *International Journal of Geographical Information Science*, 4:369–383, 1990.

[199] W. Wu, J. Mi, and W. Zhang. Generalized fuzzy rough sets. *Fuzzy sets and systems*, 151:263-282, 2003.

[200] W. N. Xiang. A gis method for riparian water quality buffer generation. *International Journal of Geographical Information Science*, 7:57–70, 1993.

[201] I. F. Yang, J. T. Lin, and C. Y. Wu. Fine structure of the compound eye of mallada basalis (neuroptera: Chrysopidae). *Annals of the Entomological Society of America*, 91(1):113–121, 1998.

[202] Y. Yang. Grey geometry. In *Proceedings of the 2004 UK Workshop on Computational Intelligence, Loughborough, UK*, pages 247–254, 2004.

[203] Y. Yang. Extended grey numbers and their operations. In *Proceedings of the 2007 IEEE International Conference on Systems, Man and Cybernetics*, pages 2181–2186, 2007.

[204] Y. Yang, D. Gillingwater, and C.J. Hinde. An intelligent system for the sustainable development of airports. In *Proceedings of The 9th World Conference on Transportation Research (WCTR)*, pages F5–02, Summer 2001.

[205] Y. Yang, D. Gillingwater, and C.J. Hinde. Applying neural networks and geographical information systems to airport noise evaluation. *Lecture Notes in Computer Science*, 3498:998–1003, 2005.

[206] Y. Yang, D. Gillingwater, and C.J. Hinde. A conceptual framework for society-oriented decision support. *AI and Society*, 19(3):279 – 291, 2005.

[207] Y. Yang, D. Gillingwater, and C.J. Hinde. Soft computing in geographical information systems. In S. Rana and J. Sharma, editors, *Chapter 2 in: Frontiers of Geographical Information technology, Springer*, volume 1, chapter 2, pages 47–62. Springer, 1 edition, 2006. ISBN 3-540-25685-7.

[208] Y. Yang, C.J. Hinde, and D. Gillingwater. A new method to evaluate a trained artificial neural network. In *Proceedings of the International Joint Conference on Neural Networks 2001 (IJCNN01)*, pages 2620–2625, Summer 2001.

[209] Y. Yang, C.J. Hinde, and D. Gillingwater. Improve neural network training using redundant structure. In *Proceedings of the International Joint Conference on Neural Networks 2003 (IJCNN03)*, pages 2023–2027, 2003.

[210] Y. Yang, C.J. Hinde, and D. Gillingwater. A new method in explaining neural network reasoning. In *Proceedings of the International Joint Conference on Neural Networks 2003 (IJCNN03)*, pages 3256–3260, 2003.

[211] Y. Yang and R. John. Grey systems and interval valued fuzzy sets. In *EUSFLAT 2003*, pages 193–197, 2003.

[212] Y. Yang, R. John, and F. Chiclana. Grey sets: A unified model for fuzzy sets and rough sets. In *Proceedings of the 2004 UK Workshop on Computational Intelligence, Loughborough, UK*, pages 239–246, 2004.

[213] Y. Yang, R. John, and F. Chiclana. Grey sets, fuzzy sets and rough sets. In *Proceedings of the 5th International Conference on Recent Advances in Soft Computing, RASC 2004, Nottingham, UK, 16-18 December 2004*, pages 348–353, 2004. (ISBN 1-84233-110-8).

[214] Y. Yang and R.I. John. Grey systems and interval valued fuzzy sets. In M. Wagenknecht and R. Hampel, editors, *Third EUSFLAT proceedings*, pages 193–197, Zittau, Germany, September 2003. European Society for Fuzzy Logic and Technology.

[215] Y. Yang and R.I. John. Roughness bounds in rough set operations. *Information Sciences*, 176:3256–3267, 2006.

[216] Y. Yang and R.I. John. Roughness bounds in set-oriented rough set operations. In *Proc. FUZZ-IEEE 2006*, pages 1461 – 1468, 2006.

[217] Y. Yang and M. Rosenbaum. Spatial data analysis with ann: Modelling to manage geoenvironmental problems caused by harbour siltation. In *the proceedings of International Symposium on Spatial Data Quality (ISSDQ'99)*, volume 1, pages 534 – 541, 18 July 1999.

[218] Y. Yang and M. Rosenbaum. Artificial neural networks linked to gis for determining sedimentology in harbours. *Journal of Petroleum Science and Engineering*, 29(3):213–220, Summer 2001.

[219] Y. Yang and M. Rosenbaum. The artificial neural network as a tool for accessing geotechnical properties. *Journal of Geotechnical and Geological Engineering*, 20(2):149 – 168, Spring 2002.

[220] Y. Yang and M. Rosenbaum. Artificial neural networks linked to gis. In M. Nikravesh, F. Amzadeh, and L.A. Zadeh, editors, *Developments in Petroleum Science, 51: Soft Computing and Intelligent Data Analysis in Oil Exploration*, chapter Chapter 28, pages 633 – 650. Elsevier Science, 2002. ISBN 3 79081 4210.

[221] Y. Yang, M. Rosenbaum, and C. Burton. An intelligent database for managing geoenvironmental change within harbours. *Environmental Geology*, 40(10):1224–1231, 2001.

[222] Y. Yang, Y. Wang, and Q. Zhang. A new method for the knowledge representation of structure diagnosis. In *Proceedings of ICCCBE-VII*, pages 1389 – 1394, Summer 1997.

[223] Y. Yang and Q. Zhang. A new method for stability analysis of underground opening using artificial neural network. *J. of Coal Science and Engineering*, 2(2):16 – 22, Autumn 1996.

[224] Y. Yang and Q. Zhang. Analysis for rock engineering using artificial neural networks. *J. of the China Railway Society*, 18(2):65 – 72, Spring 1997.

[225] Y. Yang and Q. Zhang. Analysis for the results of point load testing with artificial neural network. In *Proceedings of IACMAG'97*, pages 607 – 612, Autumn 1997.

[226] Y. Yang and Q. Zhang. A hierarchical analysis for rock engineering using artificial neural network. *Rock Mechanics and Rock Engineering*, 30(4):207 – 222, Autumn 1997.

[227] Y. Yang and Q. Zhang. Coding the interaction matrix of rock engineering system using artificial neural networks. *J. of China Civil Engineering Society*, 31(2):21 – 99, April 1998.

[228] Y. Yang and Q. Zhang. Intuitive analysis of factors for stability of rock engineering. *Chinese J. of Rock Mechanics and Engineering*, 17(3):336 – 340, Autumn 1998.

[229] Y. Yang and Q. Zhang. A new method for the application of artificial neural networks to rock engineering system. *Int. J. of Rock Mech. Min. Sci.*, 35(6):727 – 745, September 1998.

[230] Y. Y. Yao. Two views of the theory of rough sets in finite universe. *International journal of approximate reasoning*, 15(4):291–317, 1996.

[231] Y. Y. Yao. Combination of rough and fuzzy sets based on alpha level sets. In T. Y. Lin and N. Cercone, editors, *Rough sets and data mining: anbalysis for imprecise data*, pages 301–321. Kluwer Academic, Boston, MA, 1997.

[232] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[233] H. Zhang, H. Liang, and D. Liu. Two new operators in rough set theory with applications to fuzzy sets. *Information Sciences*, 166:147–165, 2004.