UNIVERSITY OF TECHNOLOGY SYDNEY

Uncertain Decision Information Processing in Warning Systems under Group Decision Making Framework

A thesis submitted for the degree of Doctor of Philosophy

By

Jun MA

June 2011

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Candidate

Acknowledgements

First, I will express my sadness that Dr. Da Ruan, my external supervisor, passed away suddenly just before the final stage of this thesis. He gave me so many generous comments and suggestions during my study; unfortunately, now I will never have the chance to present this thesis to him. May he rest in peace in heaven!

I would like to express my sincere appreciation to Dr. Guangquan Zhang and Dr. Jie Lu, my supervisors, for their encouragement and guidance. This thesis would never have been completed without their strict examination and instructive suggestions. I have learned so much from them, not only in research but also in life.

I could not have finished this thesis without the support of my family. The birth of Liwei, my son, and his crying encouraged me to struggle with piles of papers in our small but warm room. Without the love and support of my beloved wife, Ya Gao, I could not have spent many hours with my face buried behind the laptop screen. I would also like to express my heartfelt appreciation to my parents-in-law and my parents for their support throughout my life.

I would like to express my thanks to my colleagues and friends in the Decision Systems and e-Service Intelligence Lab. I cannot forget the time we have spent together in workshops and at parties. They are great people to work with.

I thank the Centre for Quantum Computation and Intelligent Systems for supporting my academic travels and providing such excellent research facilities. I thank Ms. Barbara Munday and Ms. Sue Felix for their help in proofreading and suggesting improvements to make sure that the content in this thesis is accurate.

I also thank Dr. Christopher Lucas, with the Australian Bureau of Meteorology, for providing the testing data set and instructive suggestions.

Finally, I would like to express my thanks to all staff at the School of Software and the Faculty of Engineering and Information Technology, at the University of Technology, Sydney, and the Australian Research Council. Without their technical and financial support, I would not have the chance to study in such beautiful city and wonderful research environment. It is a great honour for me.

Abstract

Human factors affect the development and deployment of an effective people-centred warning system. The study of decision information processing in a complex and dynamic decision environment can be used to handle human factors efficiently.

Taking note that group decision-making is an effective processing strategy when people make decisions in a complex and dynamic decision environment, this thesis studies four aspects of decision information processing within a group decision-making framework. The four processing aspects include 1) detecting decision information inconsistency; 2) integrating decision information; 3) predicting risk using decision information; and 4) measuring decision information similarity.

Focusing on the above four processing aspects, the thesis:

(1) Presents a rule-map technique and establishes a rule-map-based information inconsistency detection method for data inconsistency; presents a state-based domain knowledge representation technique and establishes a detection method for logical inconsistency based on this;

(2) Presents an extended physical model as an information integration framework and establishes an information integration method based on this;

(3) Presents a vector aggregation operator based on a complex fuzzy set framework and establishes an information prediction method for decision information with multiple periodic features;

(4) Presents a graduate aggregation operator and establishes a measuring method for similarities among decision information.

The thesis illustrates a decision support system prototype of decision information processing in group decision-making.

Experiments indicate that the presented techniques and methods can effectively support dynamic decision information processing in a complex decision environment.

Contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Research Background

Warning systems can help people to reduce the damage caused by man-made or natural disasters to public security and the socially sustainable development of a nation, a region, or the world. The past ten years have seen a huge number of catastrophic events, such as tsunami, earthquakes, floods, bushfires, and terrorist attacks, which have claimed hundreds of thousands of lives and cost hundreds of millions of dollars in damage and in post-event rescues and relief (a short list is given in Table 1.1). Research in 2006 showed that during the decade of 1995 to 2004, over six thousand natural disasters were recorded which resulted in more than 900,000 people dead and 2.5 billion people affected, and the cost exceeded 738 billions US dollars (Basher, 2006). These events severely threaten public safety and delay the sustainable development of society. In order to save lives and reduce the devastating impact of such events, many kinds of warning systems have been developed and deployed in natural disasters forecasting, emergency response, and national and public security fields, as well as in industries and business competition. Although these warning systems have achieved remarkable success in practice, they are still fall short of the total requirement in real applications. One important reason for this fact is that human factors have not been well-considered in developing and deploying these systems.

Human beings play an important role in developing and deploying an effective warning system and are also important components of a warning system; however, this opinion has not been commonly recognized and accepted until recent years. More

Table 1.1: A short list of natural and man-made disasters 2000 – 2011

Event	Country/Area	Date/Time	Cost (Bn US$)	Death Toll
Mumbai Terrorist Bombing	India	14/07/2011	26	
Ocean Earthquake and Tsunami	Japan	11/03/2011	309	28050
Queensland Floods	Australia	12/2010–01/2011	5.2	35
Christchurch, Earthquake	New Zealand	04/09/2010	6.5	
Haiti Earthquake	Haiti	12/01/2010	8.0	222570
Sichuan Earthquake	China	12/05/2008	85	87476
London Terrorist Bombings	UK	07/07/2005		56
Ocean Earthquake and Tsunami	Southeast Asia and East Africa	26/12/2004	1.0	184167
The 11 September Terrorist Attacks	USA	11/09/2001	44.2	2752

and more researches and investigation reports have addressed the fact that thousands of lives could be saved and the damages of such catastrophic events could be reduced if human factors were adequately considered. In recent years, implementing effective human factor processes in warning systems has drawn attention from academic communities, industrial and business fields, and the public. The United Nations (UN) released a global survey of early warning systems in 2006 and urged its member states to accelerate the deployment of people-centred warning systems following the 2004 Indian Ocean Tsunami disaster (The United Nations, 2006). In the UN's recommendation, the framework of a people-centred warning system is composed of four fundamental and closely connected components: risk knowledge, monitoring and warning service, dissemination and communication, and response capability (The United Nations, 2006). Within this framework, the engagement of people in a warning system is particularly emphasised. The people concerned are not only those who are at risk but also the operators and decision makers of a warning system (Basher, 2006). The progress of developing people-centred warning systems is currently delayed by various difficulties in technical and social aspects. One technical difficulty is how to effectively process multi-source decision information with multiple natures in a dynamic decision environment.

Decision information in a people-centred warning system typically has two fundamental forms, i.e., objective information and subjective information. Objective information has definite meanings and is epitomized by specific representations. The readings of a monitoring sensor, the signals of a detector, the flag signals, the coded message, and the output of a decision support system are all examples of objective information. Correspondingly, subjective information is often expressed by certain artificial languages or natural languages which have variable semantics for different people and in different situations. Typical examples of subjective information include an operator's awareness of an abnormal observation, a domain expert's assessment of a situation, and an emergency management authority's decision in a disastrous event. Both objective and subjective decision information have uncertainties. The uncertainty in objective information arises mainly from errors in observing procedure. The uncertainty in subjective information is mainly introduced by human beings. The differences in people's experiences when facing similar events, in their professional knowledge of relevant domains, and their personal judgement of a situation, affect their representation of an event, the appropriate interpretation, and the effective application of subjective information. Because subjective information can be misrepresented, misunderstood, and misused in a real application, processing subjective information effectively becomes an important and urgent issue in developing and deploying a high-performance people-centred warning system.

Decision information processing in a people-centred warning system has two remarkable characteristics, i.e., it is conducted dynamically, and it is group-based. The dynamic characteristics of decision information processing is deeply rooted in its application environment. Technically, a warning system refers to any system of a biological or technical nature that is deployed to inform of a future danger. It is designed for monitoring abnormal changes in the indicators of an activity, an environment, or an event to inform its users of potential risk. When an abnormal change is observed, the system should respond quickly. Because an observation is affected by various dynamically changing factors, the decision information processing system should catch up and respond to that change accordingly. This change-tracing requirement leads the processing to be conducted aggressively. Moreover, as important components of a people-centred warning system, people are responsible for developing, deploying, managing, and applying a warning system. A person's response experience, professional knowledge, and situation awareness come from his or her perception of a specific disastrous event and will affect the functionality, architecture, deployment, and application of a warning system. When processing decision information, a person must combine his or her mental model with the real situation in order to make an appropriate decision. The combination is dynamically conducted; therefore, a person can make appropriate but completely distinct decisions when he or she is confronted by two pieces of similar decision information in different situations.

Decision information processing is also group-based. A system's complexity, an environment's dynamicity, and personal cognitive limitations make it very hard, even impossible, for an individual to make an appropriate decision quickly. A well-accepted strategy to help decision-making in complex and dynamic situations is group decision making. Group decision-making can provide reference information from multiple perspectives and can balance different opinions and awareness. A big challenge in developing group decision-making techniques is the effective process of subjective opinions. As mentioned above, different people have different domain knowledge and experiences; therefore, they prefer to express their situation awareness, opinions, and judgements in their own ways. Inevitably, there is a huge amount of potential inconsistencies and undeterminable semantics in those personal expressions. When a system uses those personal expressions as decision information, much work needs to be done in order to avoid misunderstanding and misuse.

When developing a people-centred warning system, people's dynamic decisions in a complex environment are an important concern. Among the research issues related to dynamic decision-making, the efficient processing of information is a critical and fundamental issue for establishing effective dynamic decision techniques and methods. Based on the above analysis, this study takes the theory and methodology of decision information processing in a complex and dynamic decision environment within a group-based decision framework as the main research focus.

1.2 Research Objectives

This thesis has four research objectives.

Objective 1: Developing detection methods for information inconsistency

Decision information is collected from different sources and has various natures. Inevitably, the decision information used is inconsistent. This inconsistency will affect people's situation awareness and affect their ability to make appropriate decisions. To reduce the inconsistency in the decision information, this study identifies two types of information inconsistency, i.e., data inconsistency and logical inconsistency, and establishes corresponding detection methods for each of them.

Objective 2: Developing integration methods for multi-source and multi-nature information

Multi-source and multi-nature decision information is hard to use in making effective decisions. People need to understand and apply it correctly by means of information integration techniques. However, integrating decision information with multiple natures is very difficult and is a problem that has not been well-solved yet. This is particularly true when that information includes subjective opinions. This study presented a two-level information integration method to handle objective and subjective decision information simultaneously.

Objective 3: Developing prediction methods for information of multiple periodic factors

Using decision information to predict potential risk can help people to be aware in good time and make decisions in advance. Sometimes, the decision information used for prediction is a set of periodic time series. To predict risk based on multiple periodic time series is difficult and has not been studied in detail. Focusing on this problem, this study presents a method for generate effective predictions based on the information provided by multiple periodic time series.

Objective 4: Developing measuring methods for information similarity

Subjective opinions from domain experts are important types of decision information that can help users to make appropriate decisions in complex situation; however, they can also be misleading. Identifying the similarity between subjective opinions is a necessary step toward using them to advantage. A major difficulty in achieving this goal is that those opinions are small-sized and multi-natured. This study developes a measuring method to discover the similarity between subjective opinions, which can partially solve this difficulty.

1.3 Research Significance

The study will significantly contribute to dynamic decision processing in the development and deployment of people-centred warning systems.

The study focuses on four important aspects of decision information processing and presents a set of processing methods for each of them. The four processing aspects connect closely in the course of collecting decision information to dynamically supporting decision making. The methods presented can provide consistent, comprehensive, referable, and balanceable information to decision makers for obtaining better situation awareness and making appropriate decisions in complex environments. These methods are useful for designing functional modules for processing multi-source and multi-nature information when developing a people-centred warning system.

The study will significantly contribute to the research on subjective information processing.

The people-centred warning system framework emphasises the involvement and communication of human beings. People's personal situation awareness, experience, and judgement are important elements of decision information. The majority of this information has subjective features. Compared with objective decision information, subjective information is harder to represent, model, integrate, and manipulate. This study pays more attention to subjective information processing when developing the detection, integration, prediction, and measuring methods. The developed methods can be directly used to extend existing subjective information processing techniques and, in turn, improve the application of those techniques in a complex and dynamic decision environment.

The study will contribute to the research on decision-making and management.

The decision procedure in a people-centred warning system is very complicated. A big challenge in implementing effective decisions is how to develop high performance dynamic decision techniques. Humans are an important factor which affects the performance of a dynamic decision technique. Many requirements for processing human factors should be considered when developing a high performance dynamic decision technique. Among those requirements, this study particularly focuses on three, i.e., simulating group decision-making strategy, processing multi-source information, and processing subjective information. The methods developed can help organizations and individuals to employ information effectively and naturally and can support decision-making and management in a complex and dynamic situation.

1.4 Structure of the Thesis

The remaining chapters of this thesis are organized as follows.

Chapter 2 overviews related research works on information verification, information integration, information prediction, and information similarity measurement based on the four research objectives addressed in the thesis.

Chapter 3 presents definitions and properties of the concepts and terminologies used in this study. These concepts and terminologies include fuzzy sets and fuzzy numbers, aggregation operators, and complex fuzzy sets.

Chapter 4 focuses on the problem of checking information inconsistency. This study identifies two kinds of information inconsistency, i.e., data inconsistency and logical inconsistency. For checking data inconsistency, a rule-map technique is presented and a checking method based on that technique is presented. For checking logical inconsistency, a domain knowledge representation technique is established and a checking method is presented based on this technique.

Chapter 5 focuses on the problem of integrating multi-source and multi-nature information. An extended information integration framework is discussed against the backgound of the nuclear safeguards information management problem. Based on this, a two-stage information integration technique is reported.

Chapter 6 focuses on the problem of prediction warning based on multiple periodic time-series information. This study uses complex fuzzy sets to represent periodic time-series information. A product-sum aggregation operator is then developed for a set of complex fuzzy sets and is used in a prediction method for multiple periodic time-series information.

Chapter 7 focuses on the problem of measuring similarity between decision information components and their sources. This study presents a graduate aggregation algorithm and establishes a similarity measuring method based on it. The measuring method assesses the similarity of decision information at three different levels, i.e., the assessment level, the criterion level, and the decision level.

Chapter 8 describes a software prototype of decision information processing and illustrates its applications in real industrial problems.

Chapter 9 summarises the works in the thesis and highlights future works.

1.5 Selected Publications Related to the Thesis

(1) Ma, J., Zhang, G. & Lu, J. 2012, A method for multiple periodic factor prediction problems using complex fuzzy sets, *IEEE Transactions on Fuzzy Systems*, vol. 20, (in press).

(2) Ma, J., Zhang, G. & Lu, J. 2011, A dynamical analysis method of opinions in social network for decision support, in *Proceedings of IFORS 2011*, Melbourne, Australia, (abstract).

(3) Lu, J., Ma, J., Zhang, G., Zhu, Y., Zeng, X. & Koehl, L. 2011, Theme-based comprehensive evaluation in new product development using fuzzy hierarchical criteria group decision-making method, *IEEE Transactions on Industrial Electronics*, vol. 58, no. 6, pp. 2236–2246.

(4) Ma, J., Lu, J. & Zhang, G. 2010a, Decider: A fuzzy multi-criteria group decision support system, *Knowledge-Based Systems*, vol. 23, no. 1, pp. 23–31.

(5) Ma, J., Lu, J. & Zhang, G. 2010b, Team situation awareness measure using semantic utility functions for supporting dynamic decision-making, *Soft Computing – A Fusion of Foundations, Methodologies and Applications*, vol. 14, no. 2, pp. 1305–1316.

(6) Ma, J., Zhang, G. & Lu, J. 2010c, A state-based knowledge representation approach for information logical inconsistency detection in warning systems, *Knowledge-Based Systems*, vol. 23, pp. 125–131.

(7) Kuo, F., Zhou, Z., Ma, J. & Zhang, G. 2010, Metamorphic testing of decision support systems: a case study, *IET Software*, vol. 4, no. 4, pp. 294–301.

(8) Ruan, D., Lu, J., Laes, E., Zhang, G. & Ma, J. 2010, Multi-criteria group decision support with linguistic variables in long-term scenarios for Belgian energy policy, *Journal of Universal Computer Science*, vol. 15, no. 1, pp. 103–120.

(9) Zhang, G., Dillon, T. S., Cai, K., Ma, J. & Lu, J. 2010, δ-equalities of complex fuzzy relations, in *Proceedings of International Conference on Advanced Information Networking and Applications (AINA 2010)*, Perth, Australia, pp. 1218–1224.

(10) Zhang, G., Ma, J. & Lu, J. 2009b, Emergency management evaluation by a fuzzy multi-criteria group decision support system, *Stochastic Environmental Research and Risk Assessments*, vol. 23, no. 4, pp. 517–527.

(11) Ma, J., Lu, J. & Zhang, G. 2009a, Information inconsistency detection using a rule-map technique, *Expert Systems with Applications*, vol. 36, no. 10, pp. 12510–12519.

(12) Ma, J., Ruan, D., Zhang, G. & Lu, J. 2009b, Impute missing assessments by option clustering in multi-criteria group decision making problems, in *Proceedings of 2009 International Fuzzy Systems Association World Congress and 2009 European Society for Fuzzy Logic and Technology Conference*, Lisbon, Portugal.

(13) Ma, J., Zhang, G. & Lu, J. 2009c, An approximate reasoning based linguistic multi-criteria group decision-making method, in Y. Chen, V. Novák & H. Tao (eds), *Proceedings of 2009 International Conference on Quantitative Logic and Quantification of Software*, Shanghai, China, pp. 145–156.

(14) Lu, J., Zhu, Y., Zeng, X., Koehl, L., Ma, J. & Zhang, G. 2009, A linguistic

multi-criteria group decision support system for fabric hand evaluation, *Fuzzy Optimization and Decision Making*, vol. 8, no. 4, pp. 395–413.

(15) Zhang, G., Dillon, T. S., Cai, K., Ma, J. & Lu, J. 2009a, Operation properties and δ-equalities of complex fuzzy sets, *International Journal of Approximate Reasoning*, vol. 50, no. 8, pp. 1227–1249.

Chapter 2 Literature Review

This chapter briefly reviews current research progress in the fields related to the four objectives identified in Section 1.2. Because the four objectives are about four closely connected but different aspects of decision information processing, giving an overall review which covers all four aspects but does not distinguish their usage may not be the best choice. Hence, this chapter reviews related works according to their main usage in this study. Considering the fact that decision information processing is group-based, this chapter takes group decision-making as a underlying framework when selecting and reviewing related work.

This chapter is organised as follows. Section 2.1 overviews the main techniques for subjective information processing in a group decision-making framework. Section 2.2 gives an overview about information inconsistency detection. In Section 2.3, we summarise the main techniques in the integration of multi-source and multi-nature information. Section 2.4 summarises work on prediction with multiple periodic factor information. Section 2.5 gives a survey of similarity measuring methods for subjective information.

2.1 Group Decision-Making

Decision-making is a ubiquitous problem which individuals or organizations must face every day, such as “shall I bring the umbrella today” or “which business strategy should we take in the market downturn?” A decision is an action or an opinion of choice (Triantaphyllou, 2000) and decision-making is a procedure of a series of information processes. Simple decisions can be formulated by an individual decision-maker depending on personal experience, professional knowledge, and a specific method. However, complex decisions can seldom be made by individuals for several reasons: a decision problem may consider a great number of objects; the relationships between objects are complicated and indescribable; and the candidate options are difficult to evaluate. When a decision problem's complexity exceeds what an individual decision-maker can handle, multiple decision-makers are involved and group decision-making is adopted.

Group decision-making (GDM) is the process of achieving a choice or a solution for a decision problem based on the inputs and feedback of multiple decision-makers. A group's satisfactory solution (final decision) is the one that is most acceptable to all group members. GDM methods have been applied to product evaluation (Lu *et al.*, 2011), and development policy selection, as well as to social management (Munda, 2009). A typical GDM problem is composed of a set of alternatives (candidate options), a set of criteria, and a set of decision-makers. Three main steps are taken to obtain the final decision, i.e., determining the relevant criteria and alternatives, evaluating the relative impacts of alternatives, and ranking alternatives (Triantaphyllou, 2000).

The GDM procedure is composed of a sequence of information processes which cover the procedures of collection, representation, imputation, filtration, verification, and integration. Newspapers, television, the Internet, and databases are examples of information sources for a decision problem. Because of their different natures, collected information is often expressed in different forms. These results in various uncertainty is introduced into the decision information to be used. Moreover, the existing noise in those sources and the errors in the collection procedure increase the amount of uncertainty in the decision information to be used. To obtain accurate and sufficient information for decision-making, collected information should be cleaned before using, and if necessary, missing information should be imputed through machine learning, statistics, and other methods. Finally, the processed information is integrated and applied to decision-making.

Due to the complexity of human cognition, information processing systems developed for supporting GDM always focus on two crucial features in the human decision-making process: inference and computation. In general, these two features coexist in an information processing system. Considering the natures of the input information and a final decision, this study divides the existing information processing techniques into two classes, i.e. certain information processing techniques and uncertain information processing techniques. By ‘certain information processing’, we refer to techniques that have expressible processing models, accept accurate information, and generate definitive results. For example, a calculator accepts accurate input numbers and outputs accurate results through the embedded mathematical processing unit. Information is not always so accurate; therefore, uncertain information processing techniques are used to handle uncertainty in information. In these techniques, the input information is imprecise; and the processing models are undetermined. Such techniques are widely used in group decision-making (Ahn *et al.*, 2000; Herrera *et al.*, 2002; Huynh and Nakamori, 2005; Kim and Ahn, 1999).

2.2 Information Inconsistency Detection

2.2.1 Information inconsistency detection

In the GDM information process, detecting inconsistency in information is required to ensure not only the correctness of the information employed but also the validity of following the information processing procedures.

Techniques by the name of information inconsistency detection (or verification, validation) have been widely used in databases, knowledge-based systems and other fields; however, a commonly acceptable definition of information inconsistency has not yet emerged. The majority of existing techniques refer to information inconsistency as information partiality (Ahn *et al.*, 2000), conflict (Motro and Anokhin, 2006), redundancy, circularity, or deficiency (Murray and Tanniru, 1991; Preece and Shinghal, 1994; Zhang and Luqi, 1999). These definitions are mainly proposed from specific application domains such as information integration and fusion in databases and knowledge-base systems.

From the viewpoint of information integration, information inconsistency is typically divided into two levels, i.e., the inconsistency at the intentional level and the inconsistency at the extensional level (Motro and Anokhin, 2006). Intentional-level inconsistency comprises the information conflicts caused by the difference among information sources in the applied data models and the difference among information representation types; while the extensional-level inconsistency consists factual discrepancies among the sources in data values that describe the same objects.

Based on its nature and manifestation, information inconsistency is classified into data inconsistency and logical inconsistency in this study. Data inconsistency is usually manifested in the errors or the inaccuracy of a certain fact. An example of data inconsistency is in the assertion: “Sydney, the capital of Australia”. Data inconsistency is recognized by confliction with a reference standard. Logical inconsistency is not easily recognized in terms of a single fact or an isolated assertion; however, it can be disclosed through deducing the paradoxes from several seemingly correct facts. For instance, suppose the two pieces of information obtained are “A veteran of World War I died in 2006” and “The veteran was 95 years old when he died”. Each piece of information seems a rational fact. However, putting them together, we will draw an absurd conclusion that “the veteran took part in World War I when he was an infant”. This is a typical case of logical inconsistency in information in a real application.

Inconsistency exists in both static and dynamic information. Static information refers to information which has been collected and stored long before an observation time slot; in other words, this study relates static information to historical data, historical records, or knowledge in a database or knowledge base. By contrast, dynamic information, also called real-time observation, is received or observed just before an observation time slot. This study mainly focuses on information inconsistency detection in dynamic information.

Information inconsistency that potentially exists in real-time observations plays two different roles in supporting dynamic decision making. On the one hand, the inconsistency affects decision-makers' situation awareness; on the other, the inconsistency provides valuable clues regarding situation change and can help decision-makers to trace the change and make corresponding responses. Although a number of information inconsistency detection techniques have been developed, the majority of these techniques focus on static information rather than dynamic information. Quality information inconsistency detection techniques are still demanded in dynamic decision-making situations, such as emergency response (Lindell, 2000).

2.2.2 Data inconsistency detection

Detection techniques for data inconsistency have been reported in natural disaster warning (Larsen and Yager, 2000; Kuchar and Yang, 2000), power plant fault diagnosis (Gentil *et al.*, 2004), process fault detection and diagnosis (Németh *et al.*, 2007; Maurya *et al.*, 2007; Venkatasubramanian *et al.*, 2003c,a,b), sensor validation (Bass, 2000; Ibargüengoytia, 2001), database information integration (Scotney and McClean, 2003; Dawyndt *et al.*, 2005), information fusion (Kumar *et al.*, 2006), knowledge-based systems (Castro and Zurita, 1998) and terrorist attack surveillance (Mileti and Sorensen, 1990). These techniques belong to three main groups, namely, data-driven techniques, model-driven techniques, and hybrid techniques (Venkatasubramanian *et al.*, 2003c,a,b). Generally speaking, a reference standard is required to detect the data inconsistency and is set based on prior knowledge, historical data or a structure-specific process model. Determining a reference standard is relatively easy for static information. However, it is difficult for dynamic information for various reasons: 1) the prior knowledge for a dynamic decision-making situation is always incomplete and may not even be available; 2) the historical data for a dynamic decision making situation may be insufficient or have a large amount of misleading data; and 3) the functional pattern of a dynamic decision-making situation may change over time, which is difficult to be tracked (Tan *et al.*, 2007).

2.2.3 Logical inconsistency detection

Information logical inconsistency arises for many reasons (Nguyen, 2005) such as inappropriate gathering time, incorrect collecting technique, and scattered sources. Logical inconsistency is normally studied at two levels: a syntactic level and a semantic level (Nguyen, 2005). At the syntactic level, each piece of information is treated as a logical formula and any logical inconsistency deduced from these logical formulae is described as a paradox. The task of detecting logical inconsistency at the syntactic level can be conducted by showing that there is no model for those logical formulae. At the semantic level, each piece of information is linked to a specific context and is interpreted with a specific fact. Therefore, the logical inconsistency is recognisable by inferring unacceptable facts in that context. Corresponding to these two levels, detection methods are reported using fuzzy sets, matrix (Botten, 1992), binary diagrams (Mues and Vanthienen, 2004a) and Petri nets (Yang *et al.*, 2003), as well as resolution by unification (Wu and Su, 1993; Zhang and Luqi, 1999).

Information logical inconsistency deduction methods are widely used to deal with logical inconsistencies at the syntactic level (Amgoud and Kaci, 2007; de Amo and Pais, 2007; Polat, 1993; Mazure *et al.*, 1997; Wu and Su, 1993; Zhang and Luqi, 1999). These methods are mainly established on certain logic systems and their corresponding reasoning mechanisms. Hence, these methods are reasoning-based. For example, weakly-negative logic, four-valued logic and quasi-logical logic are used to implement logical inconsistency detection (Hunter, 1998, 2003). Since a logical inconsistency at the syntactic level is displayed when there is no model for a set of formulae, resolution strategies for the satisfiability problem (Chang and Lee, 1973; Russell and Norvig, 1995) are naturally introduced to detect logical inconsistency. For instance, Polat (1993) applied a unification strategy for logical inconsistency, and Mazure *et al.* (1997) used the bounded resolution technique and the local searching method for detecting inconsistency in non-monotonic knowledge bases. In general, first-order predicate logic is the most used model for analyzing, describing and detecting inconsistency, redundancy, circularity, and incompleteness (Zhang and Luqi, 1999).

Information logical inconsistency detection methods at the semantic level are mainly developed on the basis of the Petri nets (Yang *et al.*, 2003), binary directed graph (Mues and Vanthienen, 2004a), and their extensions. These methods take the objects and their relationships in a real world into account; and try to find inconsistency in them by searching conflicting facts along possible search paths. These methods are well-known as graph-based. For example, Park and Seong (2002) reported a knowledge base detection method based on extended colored Petri nets and used this method in nuclear power plant dynamic alarms analysis. Yang *et al.* (2003) also proposed a high-level Petri nets formalization model for detecting rule bases, in which each rule is represented by a Horn clause. Furthermore, Botten (1992) used a matrix to describe the rules in a knowledge base, which is similar to the incident matrix in Petri nets theory. Similarly, (Mues and Vanthienen, 2004a,b) developed a logical consistency detection method by applying binary decision diagrams.

Both the reasoning-based and graph-based methods have drawbacks when applied to logical inconsistency detection. These methods mainly focus on logical inconsistency in stored static information (Botten, 1992; Mazure *et al.*, 1997; Mues and Vanthienen, 2004a; Scarpelli and Gomide, 1994; Wu and Su, 1993) and very few apply for dynamic information. Moreover, they lack the ability to identify those models that only exist in theory. Therefore, they search all possible facts no matter whether they are meaningful or not. For example, suppose $\{A \rightarrow B, B \rightarrow C, C \rightarrow \neg A\}$ is a set of information. Clearly, $A = 0$, $B = 1$ and $C = 1$ is a model for these three rules in a general propositional logic. Hence, $A \rightarrow \neg A$ as a logical consequence of $A \rightarrow B$, $B \rightarrow C$, and $C \rightarrow \neg A$, can be deduced without difficulty. However, this conclusion is not acceptable in a real situation. The main reasons for these drawbacks are that: 1) current methods are usually based on well-defined knowledge representation techniques for general purpose knowledge rather than for domain-specific knowledge; 2) these methods use two-valued logic as the logic basis, which is unsuitable for a domain-specific situation where the used knowledge is domain-specified and the underlying logic is multiple-valued; and 3) these methods mainly focus on stored knowledge (static information) rather than real-time observations (dynamic information). Stored knowledge is static information, while real-time observations are dynamic information. In the real world, both static information and dynamic information are needed in a given processing period (Polat, 1993). In particular, when applying an application system in an uncertain and changeable situation, the processing for dynamic information is more crucial than it is for static information.

2.3 Information Integration

Information integration is an older research field (Ziegler and Dittrich, 2004). In different research areas, it has different names. Information integration is also known as data fusion and information fusion. In the research field of group decision making, information integration mostly refers to the aggregation of opinions of multiple decision-makers to generate an overall opinion. Due to the differences in professional background, experience, and personal preference, different decision-makers provide different opinions in their own ways. How to generate a trade-off of those opinions is the main task of information integration in GDM decision information processing.

2.3.1 Aggregation operators in GDM

In the GDM field, information integration is mainly conducted through aggregation operators (Delgado *et al.*, 2001; Herrera *et al.*, 2001; Herrera-Viedma *et al.*, 2004). An aggregation operator is essentially a kind of multi-variable function which projects a set of inputs in a given domain to an output in the same domain. To meet the requirements in real applications, an aggregation should have some expected properties which include the boundary condition, monotonousness, and idempotent condition sometimes (Calvo *et al.*, 2002b). Because the majority of existing aggregation operators satisfy the boundary condition, aggregation operators are known as averaging operators and are used to generate a trade-off between different opinions in GDM applications.

Commonly-used aggregation operators are roughly divided into three classes based on the underlying ideas of presenting and using them. The first type explicitly distinguishes the differences in importance among inputs and implements a kind of averaging computation. Hence, this study refers to them as average-based aggregation operators. Typical average-based aggregation operators are the arithmetic mean, the weighted arithmetic mean (Bustince *et al.*, 2008; Calvo *et al.*, 2002b), quasi-arithmetic means (Calvo *et al.*, 2002b), and OWA operators (Yager, 1993, 1994, 2004). Generally, average-based aggregation operators have a well-designed computing method of finding an average (trade-off) value in the given domain. However, some computing methods may be questioned in real applications because they are designed without considering the real requirements in application domains. The second type of aggregation operator is based on a certain integral theory which includes the Lebesgue integral, the Choquet integral (Mesiar *et al.*, 2008a,b; Meyer and Roubens, 2005, 2006), and the Sugeno integral (Marichal, 2000, 2001). These kinds of aggregation operators can be called integral-based aggregation operators. Integral-based aggregation operators are the generalized forms of some average-based aggregation operators. They are also established on certain computing methods. Unlike the average-based aggregation operators, the integral-based aggregation operators are used for continuous functions. Apart from the above two types, which are mainly based on computations, a third type is established on the basis of certain kinds of norms (e.g., the triangular norms, the uninorms (Yager and Rybalov, 1996) and the nullnorms). Seen from the viewpoint of their computing methods, norms are special binary operations. However, the most important feature of norms is their logical property. In classical and non-classical logics, norms are used as the computing methods of truth-values of logical formulae. In this sense, norm-based aggregation operators potentially have the capability to simulate human inference.

Currently existing aggregation operators often require that a participant provides a complete evaluation report; in other words, they do not consider and process cases with missing evaluations. However, a real decision problem is more or less faced with missing values. How to process missing values is, therefore, a key concern when applying an aggregation operator, but this issue has not yet been solved. Although so many powerful aggregation operators have been presented, little is known about how to select an appropriate one in real applications. Beliakov and Warren (2001) reported a solution for this problem by using a mathematical programming technique. In heir solution, an aggregation operator's form is fixed but its parameters are tuneable.

2.3.2 Linguistic methods in GDM

Regarding the uncertainty in the information process, linguistic methods are adopted in GDM problems for subjective information integration. Linguistic methods are an implementation of the concept of "computing with words (CWW)". The principal aim of CWW as introduced by Zadeh (1996) is to effectively employ natural language to approximate and automate reasoning in information and knowledge processing. In contrast to computing with numbers, CWW requires a formal representation framework, which is based on a restricted subset of natural language, as the basis for information and knowledge processing (Lawry, 2008). Since the CWW concept was first presented, various linguistic information process models have been developed and applied in many fields, such as group decision-making (Delgado *et al.*, 1998; Herrera and Martínez, 2000; Ma *et al.*, 2007), automata (Ying, 2002), information fusion (Yager, 2005), and industrial applications (Zhang and Lu, 2009). Among those application fields, GDM is a typical domain in which decision-makers' opinions on available alternatives are always expressed by natural languages.

Fuzzy-set-based CWW frameworks are extensively studied because fuzzy set provides a reasonable representation of linguistic information. Linguistic variable (Zadeh, 1975a,b, 1976) is the first framework in this technique. In this framework, a set of linguistic terms is initially identified and other terms are generated recursively from the initial linguistic terms by using logical connectives (such as "$\vee$, $\wedge$, $\neg$, $\rightarrow$") and linguistic hedges (such as "quite, very, almost"). The semantics of those linguistic terms is measured through fuzzy operations assigned to the connectives and linguistic hedges. For example, the semantic of linguistic hedge "very" on a linguistic term t is calculated by

$$S(\text{very}(t)) = S^2(t). \tag{2.1}$$

Note that some logical connectives are binary operators; the norms discussed in Section 2.3.1 can be used as computing models for these logical connectives. Another fuzzy-set-based CWW framework is the perceptual computer (Per-C) architecture (Mendel, 2002) which uses type-2 fuzzy sets as the expression of words. In Per-C, words are first encoded into type-2 fuzzy sets; then a CWW engine embedded with IF-THEN rules processes these type-2 fuzzy sets and generates new type-2 fuzzy sets as output; finally, type-2 fuzzy sets as output will be decoded to words (Mendel, 2007a,b,c). In contrast to the linguistic variable framework, Lawry (2008, 2001, 2004) presented the label semantics framework. This framework investigates the semantics of linguistic terms based on the context in which a population of agents communicate and interact. The linguistic terms in this framework are generated by using standard logic connectives; however, their semantics is measured through the appropriateness of a term and the probability mass function which indicates an agent's belief about the subset of terms used to describe an object. Linguistic hedges are an important component of natural language. Hedge algebra (Ho and Wechler, 1990, 1992; Ho *et al.*, 1999; Ho and Nam, 2002) is a mathematical framework for domain linguistic variables, in which linguistic terms are considered as an algebraic structure, such as a distributive lattice. Hedge can be seen as an operation on a linguistic term and the order of linguistics terms is established by semantic analysis. Comparing the frameworks based on linguistic variables, label semantics, and hedge algebra, it can be observed that the first two frameworks have better computation features but less inference flexibility than the latter. Hence, a combination of computation and inference frameworks is required which has not yet been achieved.

Many linguistic methods have been developed for GDM problems. A family of linguistic methods is established on a computational model. These methods determine a set of linguistic terms in advance which are used by decision-makers as their assessments on available alternatives. A computational model then takes the indexes of assessments as well as the weights of decision-makers as input and generates a number as output. If required, a term will be retrieved based on the number. Another family of linguistic methods is based on symbolic computational models. The 2-tuple model is a typical one (Herrera *et al.*, 2002; Herrera and Martínez, 2000). In this model, linguistic terms are symmetrically placed on an ordinal scale. Each linguistic term is expressed by a pair (s, a) where s is a linguistic term and a is a numeric value representing a symbolic translation. Although they are successfully applied in many fields, computation based linguistic methods still cannot fully fulfill the demand of the GDM information process because human decision-making employs much more inference than computation. Another drawback is that the obtained result is hard to interpret in a natural way because the result seldom has a corresponding linguistic term in a computation model.

2.4 Information Prediction

Prediction problems are ubiquitous. Researches are conducted in mathematics, statistics, social management, and engineering applications. This section particularly focuses on research into generating predictions from information of multiple periodic changed factors.

2.4.1 Multi-sensor data fusion

Multi-sensor data fusion is "the theory, techniques and tools which are used for combining sensor data, or data derived from sensory data, into a common representation format" (Mitchell, 2007) to "achieve improved accuracies and more specific inferences" (Hall and Llinas, 1997). It is widely used in military and civilian applications (Hall and Llinas, 1997). The study of multi-sensor data fusion crosses multiple disciplines such as mathematics, engineering, and decision theory. Researchers have presented models to describe and classify a multi-sensor data fusion system from different points of view (Mitchell, 2007; Esteban *et al.*, 2005). Among these models, the JDL data fusion model developed by the Joint Directors of Laboratories Data Fusion Working Group is the most cited (Steinberg *et al.*, 1999; Llinas *et al.*, 2004). It is a functionality oriented model and defines a five-level process framework, i.e., sub-object data assessment (level 0), object assessment (level 1), situation assessment (level 2), impact assessment (level 3) and process refinement (level 4). By contrast with the JDL data fusion model, Mitchell (2007) adopts a three-level statistical framework to describe a multi-sensor data fusion system, which includes a physical level, an informative level and a cognitive level. In recent years, the progress of techniques and expansion of application fields has enriched the research subjects of multi-sensor data fusion (Mitchell, 2007; Hall and Llinas, 1997; Esteban *et al.*, 2005). At the same time, new challenges have emerged.

2.4.2 Uncertain information processing

Research on representing and processing uncertain information has produced plentiful and substantial outcomes. Fuzzy sets, possibility theory, artificial neural networks, and Bayesian analysis, have been widely used (Hájek *et al.*, 1992). However, methods based on these techniques mainly focus on the semantic features but they ignore the periodic feature which a piece of decision information may have. A fuzzy set is often used to describe the semantic of a linguistic expression of an uncertain concept by a specified membership function. During the uncertain information processing, the membership function is unchanged; in other words, the semantic of the uncertain concept is unchanged. When the definition of the membership function is no longer appropriate to describe the semantic of that uncertain concept in the processing context, a new uncertain concept is then defined and a new fuzzy semantic representation is accordingly introduced (Karsak and Tolga, 2001), although the newly created concept should be the same as the old one. This strategy is easy to apply but it has some drawbacks: 1) it is inflexible (e.g., it is necessary to define many completely different fuzzy semantic representations for uncertain concepts even though they have similar semantics); and 2) it may lead to potential semantic conflict (e.g., the same fuzzy semantic representation may be used to describe two different uncertain concepts in different situations, or the same uncertain concept may have different semantic representations in different situations in the same application); 3) it cannot describe a time-dependent semantic change. Let us consider a linguistic term used to describe the weather condition. People may use the same word "cool" to describe a day's temperature irrespective of whether it is winter or summer, although a "cool day" has a different meaning semantically in summer and in winter. People can correctly distinguish the difference and seldom need to define two different terms for summer and winter. Note that the semantics change over time; hence, providing a temporal process when describing the semantics of uncertain information is possible and rational. Following on this idea, Chapter 6 reports a processing method which uses complex fuzzy set to represent information with both temporal and semantic features.

2.4.3 Complex fuzzy sets

The concept of complex fuzzy sets (CFSs) was first presented in Ramot *et al.* (2002). A CFS is a kind of extension of a traditional fuzzy set which takes the elements in the complex unit disc as the membership degrees of objects in a universe of discourse. Introducing complex numbers into the theory framework is a primary direction to extend the conventional fuzzy set. Buckley (1989); Buckley and Qu (1991); Buckley (1992)[1], Zhang (1991, 1992), Yoon (1996), Nguyen *et al.* (1998) and others (Moses *et al.*, 1999) have presented proposals on the complex-valued extension of conventional fuzzy sets. Unlike Buckley's, Yoon's and Zhang's proposals, in which complex numbers are used as elements in a universe of discourse, the complex numbers in a CFS are used as the membership degrees of elements in a universe of discourse. In other words, the membership function of a CFS is truly a complex-valued function. Before the presentation of the CFS concept, another complex-valued membership degree has been used in Nguyen *et al.* (1998). However, the introduction of a

[1] Recently, Qiu *et al.* (2009) have reported and corrected an error of the concepts of generalized fuzzy complex numbers.

complex-valued membership degree is the result of different philosophical concerns. When resolving a truth-value equation in problems related to logical paradox, Nguyen discovered that the solutions of such kinds of equations are complex-valued. Hence, introducing complex-valued membership functions arises naturally from the theoretical requirement. The complex-valued membership degrees in a CFS are used to represent observations which have both modulus and phase features in real applications, such as annual sunspot number counting and referential signal detection (Ramot *et al.*, 2002). Hence, introducing complex-valued membership functions in CFSs meets the requirements of engineering applications. Recently, Tamir *et al.* (2011) presented another interpretation of the complex-valued membership degrees from a complex fuzzy class, which is embedded in complex fuzzy set class operations.

Because of the introduction of complex-valued membership functions in CFSs, the operations on CFSs are not intuitive compared to those on conventional fuzzy sets. Ramot *et al.* (2002, 2003) presented some set-theoretical operations. These operations are mainly defined on the modulus part of the complex-valued membership degrees, rather than the phase part. In order to deliver operations that better illustrate the phase part of a CFS, Dick (2005) presented the concept of rotational invariance and defined a complex fuzzy logic system. Aghakhani and Dick (2010) developed learning algorithms based on complex fuzzy logic and applied them to two time series prediction problems. Chen *et al.* (2011) presented a neuro-fuzzy system based on complex fuzzy logic, which implements complex fuzzy rules and is applied to time-series forecasting problems. Moreover, Zhang *et al.* (2009) defined the δ-similarity of two CFSs considering both the modulus and the phase parts. Deshmukh *et al.* (2008) designed and implemented a complex fuzzy logic system. Recently, Li and Chan (2011) used CFSs to resolve an image restoration problem.

2.4.4 Multiple periodic factor prediction

This section focuses on the Multiple Periodic Factor Prediction (MPFP) Problem. In real MPFP problems, the collected information is often expressed by a set of time series (or a set of fuzzy time series). There have been many powerful time series analysis and forecasting techniques from statistics and engineering applications (De Gooijer and Hyndman, 2006; Hamilton, 1994; Kirchgässner and Wolters, 2007). Franses and Paap (2004) reported some fixed-period periodic time series processing models. Recently, Shang and Hyndman (2011) presented a non-parametric method to forecast time series with a seasonal univariate and applied it to monthly sea surface temperature data. In De Gooijer and Hyndman (2006), a time-line overview of the progress in time series forecasting is presented according to the adopted models in the cited literature. This overview indicates that: 1) most time series research is conducted on univariate rather than multivariate objects; 2) most research is conducted under the linear model framework; and 3) work on combining multiple time series is still limited (Li and Chiang, 2011).

Effectively integrating predictions of multiple periodically changing factors in an MPFP problem is an important step for generating prediction. The literature indicates that a great number of aggregation operators have been presented as information integration techniques, which can be used to support multi-criteria decision making and group decision-making. These operators cannot be directly used in solving the MPFP problem because they are mainly designed to deal with semantic uncertainty in information. An MPFP problem, however, should handle the periodicity simultaneously when processing semantics. Hence, the information integration technique for periodically changing factors is important, but as yet there is no effective solution. It is noted that some research works do consider the aspect of periodicity, but without presenting a corresponding applicable processing method. Stach *et al.* (2008) presented a fuzzy-cognitive-maps-based technique to conduct time series prediction. This technique is implemented at both a numerical and a linguistic level. Karsak and Tolga (2001) presented a four-cycle method to help managers implement an incremental and continuous information system planning process, in which periodicity is taken as an adjusting factor. Considering the diversity and complexity of periodically changing factors, Keyno *et al.* (2009) reported a method to decrease the effects of those factors in electricity demand forecasting problems. Moreover, Bichescu and Fry (2009) discussed the performance differences of several supply chain models in which order quantity and shipping frequency are taken as decision variables in a simple supply chain with a single retailer and a single supplier. Research has shown that periodicity becomes an important concern in decision problems in different areas (Ferreira and de Almeida, 2009; Fontini *et al.*, 2010; Ustun and Demirtas, 2008a,b).

2.5 Information Similarity Measurement

Opinion analysis is extensively studied in social psychology fields (Banisch and Araújo, 2010; Sigall and Aronson, 1967); recently, requirements for effectively extracting, summarizing, and segmenting opinions of general or specific users has boosted the growing research on opinion mining and sentiment analysis (Ganesan *et al.*, 2010; Liu, 2010; Pang and Lee, 2008; Vechtomova, 2010). Commonly recognized, opinion mining research belongs to the field of text analysis (Pang and Lee, 2008); therefore, currently reported methods focus mainly on how to efficiently extract and summarize opinions from texts distributed among web blogs posts (Vechtomova, 2010), bulletin board systems (BBS) (Huang *et al.*, 2008), online feedback (Hu and Liu, 2004; Wu *et al.*, 2010), and web forums (Ganesan *et al.*, 2010). Many opinion mining systems have been developed and applied (Chen and Zimbra, 2010; Ganesan *et al.*, 2010; Huang *et al.*, 2008; Pang and Lee, 2008; Wu *et al.*, 2010). In the field of multi-criteria group decision-making (MCGDM), study of opinion analysis is conducted in two main areas. Qualitative studies analyse and simulate the behaviour patterns of users based on their opinions of a considered affair (Miller and Morrison, 2009; Pandelaere *et al.*, 2010). Quantitative research focuses on how to represent and process opinions in a computational framework to support decision-making (Chakraborty and Chakraborty, 2010; Herrera *et al.*, 2001; Meyer and Roubens, 2006; Pasi and Yager, 2006). For instance, fuzzy sets and fuzzy logic are widely used as opinion representation and process facilities (Herrera-Viedma *et al.*, 2004, 2007; Huynh and Nakamori, 2005) because they can effectively interpret and model subjective information with uncertainties.

Similarity measurement is widely studied in human knowledge representation, behaviour analysis, and real-world problem solving (Yu *et al.*, 2008). A similarity measurement can be established on various theories (e.g., classical and/or fuzzy set theories, classical and/or non-classical logics (Cross and Sudkamp, 2002)) and applied to image processing (Inglada and Mercier, 2007; Yu *et al.*, 2008), natural language understanding (Iosif and Potamianos, 2010), recommender systems (Nanopoulos *et al.*, 2010), and other applications. In MCGDM, similarity measures defined on a fuzzy set are particularly useful. Because many similarity measures can be sourced from their counterparts defined on ordinary sets, research on their relationships has been conducted. For example, Wang *et al.* (1995) compared commonly-used similarity measures on the ordinary set. De Baets *et al.* (2001) systematically discussed a way of generating a similarity measure for ordinary sets and compared it with 28 other similarity measures. Recently, Bosteels and Kerre (2007) presented a family of cardinality-based fuzzy similarity measures which is specified by three parameters and De Baets *et al.* (2009) studied the transitivity of cardinality-based similarity measures. Generally speaking, a similarity measure can be induced from a distance measure. Therefore, investigating the relationship between them is very important (Balopoulos *et al.*, 2007). The majority of existing similarity measures are defined on the Euclidean space and the ultimate measurement is a crisp value. Noting that a crisp value cannot sufficiently depict the fuzziness in real cases, Chakraborty and Chakraborty (2010) defined a similarity, whose measurement is a fuzzy set. Using this distance, they implemented a clustering algorithm to solve a group decision-making problem.

Chapter 3 Preliminaries

This study focuses on decision information processing in complex and dynamic decision environment. The decision information is collected from multiple sources and is with multiple natures. To effectively process decision information, this study presents a set of processing methods and techniques. This chapter briefly introduces the terminologies and definitions used in the study.

3.1 Fuzzy sets and fuzzy numbers

The concept of fuzzy set was originally presented by Zadeh (1965). Since it was been presented, theories, methods, and techniques based fuzzy sets have been widely used in many application fields. Formally, a fuzzy set is defined as below.

Definition 3.1.1 (fuzzy set)**.** *(Klir and Yuan, 1995) Let X be a universal set. A fuzzy set A of X is defined by a membership function μ_A:*

$$\mu_A : X \to [0, 1] \tag{3.1}$$

which assigns each element x in X a real number $\mu_A(x)$.

The real number $\mu_A(x)$ is called the degree of membership (or membership degree) of x in A.

Remark 3.1.1. *Definition 3.1.1 gives the commonly-used definition of a fuzzy set. However, it is not a strict definition. Essentially, Definition 3.1.1 defines a fuzzy subset rather than a fuzzy set.*

In algebraic theory, the unit interval $[0, 1]$ of real number is a special totally-ordering lattice. A generalized definition of fuzzy set is given in terms of a lattice as below.

Definition 3.1.2 (lattice-valued fuzzy set)**.** *(Zhang, 1994) Let X be a universal set and L be a lattice. A lattice-valued fuzzy set A of X is defined by a lattice-valued membership function μ_A:*

$$\mu_A : X \to L \tag{3.2}$$

which assigns each element x in X an element $\mu_A(x)$ in L.

Definition 3.1.3. *Let A be a fuzzy set of X. The α-cut of a fuzzy set A is a subset of X such that*

$$A_\alpha = \{x \in X \mid \mu_A(x) \geqslant \alpha\}. \tag{3.3}$$

The strict α-cut of A is a subset of X such that

$$A_{\alpha+} = \{x \in X \mid \mu_A(x) > \alpha\}. \tag{3.4}$$

When $\alpha = 0$, the strict 0-cut A_{0+} is called the support set of A.

In the following, we will denote the α-cut of a fuzzy set A by $[A^L_\alpha, A^R_\alpha]$.

Theorem 3.1.1. *Let A be a fuzzy set of X. Then A can be represented by its α-cuts as below:*

$$A = \bigcup_{\alpha \in [0,1]} A_\alpha. \tag{3.5}$$

■

Definition 3.1.4 (normal fuzzy set)**.** *Let A be a fuzzy set of X. A is called a normal fuzzy set if A_1 is non-empty.*

In the following, the set of all fuzzy sets of X is denoted by $\mathscr{F}(X)$ and the set of all lattice-valued fuzzy sets of X is denoted by $\mathscr{L}(X)$.

The set of real numbers $\mathbb{R}$ and the set of complex numbers $\mathbb{C}$ are two commonly used universal sets for defining a fuzzy set. When the set $\mathbb{R}$ is used as the universal set of a fuzzy set, a special fuzzy set called fuzzy number is defined.

Definition 3.1.5 (fuzzy number)**.** *Let A be a fuzzy set of $\mathbb{R}$. A is called a fuzzy number if A satisfies the following conditions:*

(1) A is a normal fuzzy set;

(2) A_α is a closed interval for any $\alpha \in [0, 1]$; and

(3) A_{0+} is bounded.

In real applications, fuzzy numbers in the following shapes are commonly used.

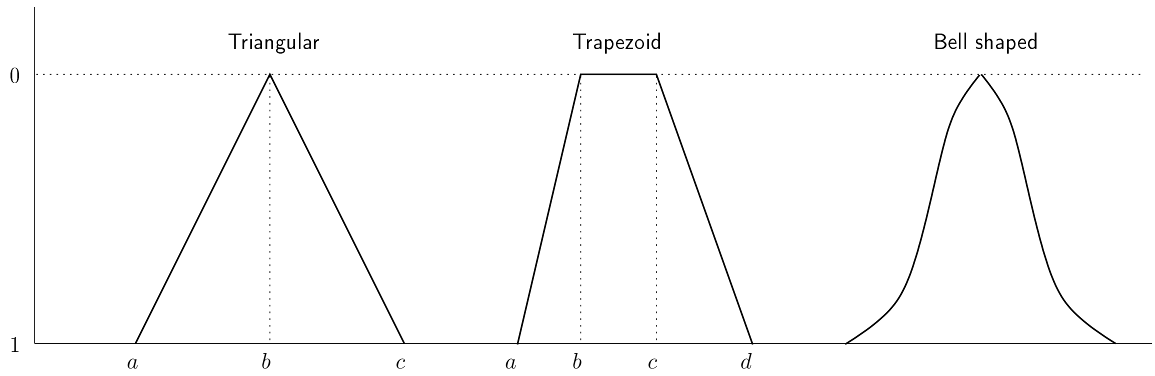


Figure 3.1: Shape of commonly-used fuzzy numbers

Definition 3.1.6 (triangular fuzzy number)**.** *A triangular fuzzy number A is denoted by a triplet (a, b, c) and has membership function μ_A as below:*

$$\mu_A(x) = \begin{cases} 0 & x < a; \\ \frac{x-a}{b-a} & a \leqslant x < b; \\ \frac{x-c}{b-c} & b \leqslant x \leqslant c; \\ 0 & x > c. \end{cases} \tag{3.6}$$

When $[a, c]$ is a subset of the unit interval of real numbers, the triangular fuzzy number (a, b, c) is also called a normalized positive fuzzy number.

For real numbers, we have four basic arithmetic operations. Similarly, we can define the four basic arithmetic operations on fuzzy numbers (Klir and Yuan, 1995).

Let A and B be two fuzzy numbers. We define the four basic arithmetic operations on their α-cuts as

$$A_\alpha + B_\alpha = \left[A_\alpha^L + B_\alpha^L, A_\alpha^R + B_\alpha^R\right]$$
$$A_\alpha - B_\alpha = \left[A_\alpha^L - B_\alpha^R, A_\alpha^R - B_\alpha^L\right]$$
$$A_\alpha \times B_\alpha = \left[\min\left(A_\alpha^L B_\alpha^L, A_\alpha^L B_\alpha^R, A_\alpha^R B_\alpha^L, A_\alpha^R B_\alpha^R\right), \max\left(A_\alpha^L B_\alpha^L, A_\alpha^L B_\alpha^R, A_\alpha^R B_\alpha^L, A_\alpha^R B_\alpha^R\right)\right]$$

and if $0 \notin B_\alpha$,

$$\begin{aligned} A_\alpha / B_\alpha &= \left[A_\alpha^L, A_\alpha^R\right] \times \left[1/B_\alpha^R, 1/B_\alpha^L\right] \\ &= \left[\min\left(\frac{A_\alpha^L}{B_\alpha^L}, \frac{A_\alpha^L}{B_\alpha^R}, \frac{A_\alpha^R}{B_\alpha^L}, \frac{A_\alpha^R}{B_\alpha^R}\right), \max\left(\frac{A_\alpha^L}{B_\alpha^L}, \frac{A_\alpha^L}{B_\alpha^R}, \frac{A_\alpha^R}{B_\alpha^L}, \frac{A_\alpha^R}{B_\alpha^R}\right)\right] \end{aligned}$$

Suppose $\circ$ is an operator of "$+$, $-$, $\times$, $/$", then

$$(A \circ B)_\alpha = A_\alpha \circ B_\alpha \tag{3.7}$$

where $\alpha \in [0, 1]$.

3.2 Aggregation operators

Aggregation (fusion) of several input values into a single output value is an indispensable tool not only of mathematics or physics, but of a majority of engineering, economical, social and other sciences (Calvo *et al.*, 2002a).

Definition 3.2.1 (aggregation operator)**.** *Let X be a universal set. An n-ary aggrega-*

tion operator $\mathbf{A}_{(n)}$ *on* X *is a mapping*

$$\mathbf{A}_{(n)} : X^n \to X. \tag{3.8}$$

Remark 3.2.1. *In many applications the set* X *is often a closed interval of real numbers. In group decision-making, the set* X *is often the set of linguistic terms used or the set of indexes of those terms.*

In general, the number of the input values to be aggregated is not known. Therefore an aggregation operator $\mathbf{A}$ should be a mapping

$$\mathbf{A} : \bigcup_{n\in\mathbb{N}} X^n \to X. \tag{3.9}$$

Thus $\mathbf{A}|_{[0,1]^n} = \mathbf{A}_{(n)}$ for all $n \in \mathbb{N}$.

Below are typical examples of aggregation operators:

Example 3.2.1 (arithmetic mean)**.** *Let* X *be the a closed interval of real numbers. The arithmetic mean is defined as*

$$\mathbf{M}(x_1, x_2, \ldots, x_n) = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{3.10}$$

Example 3.2.2 (min-type weighted aggregation)**.** *(Ruan et al., 2003)*

$$\min(x_1, x_2, \ldots, x_n) = \min\left(g(w_1, x_1), g(w_2, x_2), \ldots, g(w_n, x_n)\right);$$

where $g\ : H \times H \to H$ *is a weight transformation function on a finite ordered set* H*.*

Example 3.2.3 (max-type weighted aggregation)**.** *(Ruan et al., 2003)*

$$\max(x_1, x_2, \ldots, x_n) = \max\left(g(w_1, x_1), g(w_2, x_2), \ldots, g(w_n, x_n)\right); \tag{3.11}$$

where $g : H \times H \to H$ *is a weight transformation function on a finite ordered set* H.

Example 3.2.4 (Yager's ordered weighted averaging (OWA) aggregation)**.** *(Yager, 2004)*

$$\mathbf{OWA}(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{n} w_j b_j,$$

here, b_j *is the* j*-th largest of* x_i*,* $i = 1, \ldots, n$*.*

Example 3.2.5 (weighted sum)**.**

$$\mathbf{WS}(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{n} w_j x_j, \quad w_j \geqslant 0, \sum_{j=1}^{n} w_j = 1.$$

Moreover, the fuzzy integral (e.g., the Choquet integral and the Sugeno integral) can also be used as aggregation operators (Marichal, 2001, 2000).

An aggregation operator needs to satisfy some properties to meet the requirements of real applications. Typical properties are listed below.

Definition 3.2.2 (idempotent)**.** *Let* $\mathbf{A} : \bigcup_{n\in\mathbb{N}}[0,1]^n \to [0,1]$ *be an aggregation operator. An element* $x \in [0,1]$ *is called* $(\mathbf{A})$*)-idempotent element whenever* $\mathbf{A}_{(n)}(x, \ldots, x) = x$ *for all* $n \in \mathbb{N}$*.* $\mathbf{A}$ *is called an idempotent aggregation operator if each* $x \in [0,1]$ *is an idempotent element of* $\mathbf{A}$*.*

Definition 3.2.3 (Continuity)**.** *An aggregation operator* $\mathbf{A} : \bigcup_{n\in\mathbb{N}}[0,1]^n \to [0,1]$ *is called a continuous aggregation operator if for all* $n \in \mathbb{N}$ *the operator* $\mathbf{A}_{(n)} : [0,1]^n \to [0,1]$ *is continuous, that is if*

$$\forall x_1, \ldots, x_n \in [0,1], \forall (x_{1j})_{j\in\mathbb{N}}, \ldots, (x_{nj})_{j\in\mathbb{N}} \in [0,1]^n : \lim_{j\to\infty} x_{ij} = x_i \tag{3.12}$$

for $i = 1, \ldots, n$*, then*

$$\lim_{j\to\infty} \mathbf{A}_{(n)}(x_{1j}, \ldots, x_{nj}) = \mathbf{A}_{(n)}(x_1, \ldots, x_n). \tag{3.13}$$

In engineering applications continuous aggregation operators are usually applied, reflecting the property that a “small” error in inputs cannot cause a “big” error in the output. From the mathemtical point of view, because of the compactness of domains $[0,1]^n$, $n \in \mathbb{N}$, the continuity of an aggregation operator $\mathbf{A}$ is equivalent to its uniform continuity expressed by

$$\begin{gathered}\forall \epsilon > 0, \forall n \in \mathbb{N}, \exists \delta > 0 : |x_i - y_i| < \delta, i = 1, \ldots, n \\ \Downarrow \\ |\mathbf{A}(x_1, \ldots, x_n) - \mathbf{A}(y_1, \ldots, y_n)| < \epsilon\end{gathered} \tag{3.14}$$

Definition 3.2.4 (Intermediate value property)**.** *Let* $(x_1, \ldots, x_n)$, $(y_1, \ldots, y_n) \in [0,1]^n$, $n \in \mathbb{N}$ *be any* n*-tuples such taht* $x_i \leqslant y_i$, $i = 1, \ldots, n$. *An aggregation operator* $\mathbf{A}$ *has the intermediate value property if* $\forall z \in [\mathbf{A}(x_1, \ldots, x_n), \mathbf{A}(y_1, \ldots, y_n)]$, $\exists z_i \in [x_i, y_i]$, $i = 1, \ldots, n$, *such that*

$$\mathbf{A}(z_1, \ldots, z_n) = z. \tag{3.15}$$

An important analytical property of functions of n variables allowing to estimate the error when dealing with imprecise input data is the Lipschitz property. Recall that an aggregation operator $\mathbf{A}$ fulfills the Lipschitz property with constant $c \in (0, \infty)$ ($\mathbf{A}$ is c-Lipschitz for short) if $\forall n \in \mathbb{N}$, $\forall (x_1, \ldots, x_n)$, $(y_1, \ldots, y_n) \in [0,1]^n$,

$$|\mathbf{A}(x_1, \ldots, x_n) - \mathbf{A}(y_1, \ldots, y_n)| \leqslant c \sum_{i=1}^{n} |x_i - y_i|. \tag{3.16}$$

3.3 Implication operators

Implication operators are widely used in mathematical logic systems, which can be applied to interpret the inference relationship between an antecedent and a consequence.

An implication operator is a binary operation with two inputs and one output. The inputs and output are generally elements in the unit interval of real numbers which is used as the truth values of logical formulae. An implication operator "$\rightarrow$" needs to satisfy the boundary conditions:

$$0 \rightarrow 0 = 0 \rightarrow 1 = 1 \rightarrow 1 = 1; \quad 1 \rightarrow 0 = 0. \tag{3.17}$$

Typical implication operators include:

- Łukasiewicz:

$$a \rightarrow b = \min\{1, 1 - a + b\}.$$

- Standard strict:

$$a \rightarrow b = \begin{cases} 0 & a > b \\ 1 & a \leqslant b \end{cases}$$

- Kleene-Dienes:

$$a \rightarrow b = \max\{1 - a, b\}.$$

- Kleene-Dienes-Łukasiewicz:

$$a \rightarrow b = 1 - a + a \cdot b.$$

- Yager's implication:

$$a \rightarrow b = b^a,$$

where $a, b \in [0, 1]$.

3.4 Complex fuzzy sets

Complex fuzzy set is an extension of the conventional fuzzy set. It was presented by Ramot *et al.* (2002). Different from the conventional fuzzy set, the membership degree of an element x in the universal set X is no longer a real number but a complex number. The concept of complex fuzzy sets roots in representing periodicity and uncertainty in observations. Formally, a complex A is defined below.

Definition 3.4.1 (complex fuzzy set)**.** *Let X be a universal set and $\mathbb{C}$ is the set of complex numbers. A complex fuzzy set A over X is defined by a complex-valued membership function μ_A such that for any $x \in X$,*

$$\mu_A(x) = r(x) \cdot e^{\omega(x)}, \tag{3.18}$$

where $r(\cdot)$ is a conventional fuzzy set and $\omega(\cdot)$ is a periodic function.

In a complex fuzzy set, the function $r(\cdot)$ is reconginsed as module function and the function $\omega(\cdot)$ is a phase function. The introduction of the phase function increases the difficult to establish effecient and rational operations on complex fuzzy sets. Ramot *et al.* (2002) listed a set of properties which can be used to define operations on complex fuzzy sets. Because the concept of complex fuzzy sets was just presented recently, well-defined operations on complex fuzzy sets have not been detailed studied yet.

In existing literatures, we can meet another similar concept which is called "fuzzy complex set". A fuzzy complex set takes a set of complex numbers as the univeral set X and the membership degree of each complex number is still a real number in the unit interval of real numbers. Compared with Definition 3.4.1, a fuzzy complex set is still a conventional fuzzy set.

Chapter 4 Decision Information Inconsistency Detection

This chapter focuses on the problem of detecting information inconsistency.

4.1 Data inconsistency and logical inconsistency

Different from the two-level information inconsistency definition given by Motro and Anokhin (2006), this study divides the inconsistencies of decision information in a dynamic decision environment into two fundamental types, namely, the data inconsistency and the logical inconsistency regarding to the manifestation of them. The data inconsistency occurs when a piece of decision information is incompatible with a normal pattern. For instance, observational errors and incorrect record in a database are treated as data inconsistency. Logic inconsistency exists in a set of information which may lead to paradoxes although each piece of it seems correct. For example, exclusive observations and conflict conclusions imply logic inconsistency.

To detect these two kind of information inconsistencies, this chapter develops a rule-map technique for detecting data inconsistency and a domain knowledge representation technique for detecting logical inconsistency. Section 4.2 mainly focus on the data inconsistency and discusses the rule-map technique. Section 4.3 focuses on logical inconsistency and discusses the domain knowledge representation technique.

4.2 Data inconsistency detection

4.2.1 Data inconsistency detection problem

Formally, a data inconsistency detection problem is addressed below.

Problem 4.2.1 (Data inconsistency detection problem)**.** *Suppose S is a monitored object in a dynamic decision environment, which has a perceivable, but unclear, functional pattern. H is a time series whose elements are historical observations of S before time t. Let $O(t)$ be a reported observation at time t. The data inconsistency detection problem is to evaluate whether the reported observation $O(t)$ is consistent with the unclear functional pattern of S.*

Problem 4.2.1 illustrates a decision environment which has the following features: 1) prior knowledge about the functional pattern is not available; 2) the functional pattern exists but is unclear; and 3) the functional pattern can be perceived from historical observations. Such kind of decision environment is ubiquitous. In Problem 4.2.1, the observations of S are not necessarily assumed to be numeric values. They can be risk grades if S is an emergency event or a certain assessments if S is an optional decision.

4.2.2 A rule-map technique

4.2.2.1 Overview

A rule-map technique is established to solve Problem 4.2.1 on the assumptions: 1) the functional pattern of S can be modelled by multiple descriptive models from different perspectives; 2) the outputs of these descriptive models can be used to establish a reference standard for detecting the inconsistency of the observation $O(t)$; and 3) when some of descriptive models fail to work at the time t, the others are still applicable.

Using multiple descriptive models for detecting information inconsistency has better fitness to the dynamic decision environment than using a single mode. The performance of using a single descriptive model to estimate the observation $O(t)$ is easily influenced by the historical observations used to construct that model. Due to the uncertainties existed in observing procedure, the historical observations, collected to establish a descriptive model, are not completely reliable. Hence, the output, which is used as estimation of $O(t)$, of such a model may be not reliable also. To reduce the risk of using an unreliable output in solving Problem 4.2.1, combining outputs of multiple descriptive models is an applicable strategy. A typical application of this strategy is that we often use group decision to replace individual decision in complex decision. A benefit of using this strategy in solving Problem 4.2.1 is that when some models fail to work, other models can still be used to trace the functional pattern and generate acceptable estimation to complete the detection task.

Using multiple descriptive models can balance different observing perspectives. Each single descriptive model may be established with more concentration on some perspectives than others. Hence, it is hard to get a comprehensive image of the monitored object's functional pattern from a single description model. Using multiple descriptive models, however, can combining perspectives used in different single models to provide a comprehensive image. Moreover, a single descriptive model may not track the change of an object's functional pattern timely due to lack of adjustable parameters and fixed structure. Multiple descriptive models offer more adjustable parameters and provide both possibility and flexibility of choosing the most appropriate models for using.

The rule-map technique establishes a mechanism for selecting and using descriptive models dynamically based on their performances. Descriptive models have discrepancies in various aspects such as flexibility and accuracy. A question arises naturally from these discrepancies that how to select appropriate models at a given moment. To answer this question, we should know which models should be used in a long run and which models can only be used in a specific period. In the rule-map technique, a covering relationship between different descriptive models (see Definition 4.2.5) is defined. A descriptive model is said to cover another model over an observing period if it has better performance in that period. By means of the covering relationship, a rule-map technique organizes multiple descriptive models in a multi-level hierarchy (called a rule-map) and adjusts the hierarchy dynamically. Because the covering relationship has close linkage to the performances of descriptive models, it is used as a reference for selecting models at a given moment.

A difficulty in solving Problem 4.2.1 is how to set a reference standard for information consistency because the functional pattern is unclear. Although the reference standard is hard to define due to the complexity of a dynamic decision environment, noting that each descriptive model can generate an estimation of the real observation, the rule-map technique, therefore, combine these estimations to create the reference standard dynamically. This strategy can be guaranteed by the covering relationship between descriptive models and the dynamic adjustment of a rule-map.

4.2.2.2 Data rules and rule-map

In the rule-map technique, a descriptive model is called a data rules (see Definition 4.2.1); historical observations are called data; and a real-time observation is simply called an observation.

Without loss of generality, let X be the set of all possible observations (states or values) of a monitored object; $\mathcal{T}$ be the set of all time slots at which data and observation are collected; and $O(t)$ ($\in X$) and $E(t)$ ($\in X$) be the observation and the estimation at time slot t ($t \in \mathcal{T}$) respectively. Noted that, $O(t)$ is really collected or observed; but $E(t)$ is the output from a description model.

Definition 4.2.1. *A data rule (rule shortly) r is a triple $(m(r), g, f)$, where $m(r)$ is a non-negative integer, $g : \mathcal{T} \rightarrow \mathcal{T}^{m(r)}$ and $f : X^{m(r)} \rightarrow X$ are two maps such that $g(t) = (t_1, \cdots, t_{m(r)})$ and $E(t) = f(O(t_1), \cdots, O(t_{m(r)}))$ where $t_i < t$, $i =$*

$1, \ldots, m(r)$.

In Definition 4.2.1, g determines a set of time slots t_1, it_2, $\cdots$, $t_{m(r)}$ such that an observation $O(t)$ at time t is related to the observations at those time slots; and f clarifies how to generate an estimation from the observations at t_1, $\cdots$, $t_{m(r)}$. For example, suppose the observations form a Fibonacci Sequence. Let $m(r) = 2$, $g(t) = (t-1, t-2)$ and $f(O(t-1), O(t-2)) = O(t-1)+O(t-2)$, respectively. Then a data rule $r = (2, g, f)$ is obtained which can be used to generate estimations at any time slot t, $t \geqslant 2$. The rule r indicates that the estimation $E(t)$ is the sum of the observations at the last two nearest time slots ($t-1$ and $t-2$) before t. For convenience, we use $r(t)$ for $E(t)$ in the following.

To depict the performance of a data rule, two measurements are used, i.e. the feasible degree (see Definition 4.2.2) and the reliable degree (see Definition 4.2.3).

Let $T = \{t \mid t^* \geqslant t \geqslant t_\perp\}$ ($\subseteq \mathcal{T}$) be an observing period over which we can measure the feasible degree and reliable degree of a data rule r, where $t_\perp$ and t^* are the starting and ending time slots of the period. Also, we let $T(r) \subseteq T$ be the set of time slots at which rule r is applied, i.e. r can produce an estimation $r(t)$ for $t \in T(r)$. The feasible degree of a data rule r is defined as:

Definition 4.2.2. *The feasible degree (or feasibility) of data rule r over T is*

$$\delta(r, T) = \frac{|T(r)|}{|T|}. \tag{4.1}$$

The following proposition obviously holds.

Proposition 4.2.1. *In a given period T, if $|T(r_1)| \leqslant |T(r_2)|$, then $\delta(r_1, T) \leqslant \delta(r_2, T)$.*

■

Proposition 4.2.1 indicates that the feasible degree of a data rule r increases positively to the number of estimations it gives. However, when $T_1 \subseteq T_2$, there is no guarantee that $\delta(r, T_1) \geqslant \delta(r, T_2)$ or $\delta(r, T_1) \leqslant \delta(r, T_2)$.

The feasible degree of a data rule reflects the extent of that data rule being able to work over a given period; in other words, a data rule with higher feasible degree can produce larger number of estimations. However, the feasible degree cannot affirm that the obtained estimations are acceptable. Therefore, a second measurement, i.e., the reliable degree, is given in Definition 4.2.3 to measure to what extent the estimations are accurate.

In order to evaluate whether an estimation $r(t)$ is accurate or not, a natural way is to check the error between $r(t)$ and the real functional pattern. Because the real functional pattern is unclear, we use a reference standard to take its place. Formally, let $\varepsilon > 0$ be an acceptable error between an estimation and the reference standard. An estimation $r(t)$ is called acceptable if $d(r(t), R(t)) < \varepsilon$ where $R(t)$ is the reference standard at t and $d(x, y)$ is a given distance measurement between x and y. In a given observing period, the overall acceptability of a data rule is called the reliable degree and is defined as follows.

Definition 4.2.3. *The reliable degree (or reliability) of a data rule r under a given acceptable error scale ε in a given observing period T is*

$$\tau(r, \varepsilon) = \frac{|T(r, \varepsilon)|}{|T(r)|}. \tag{4.2}$$

where $T(r, \varepsilon) = \{t \in T(r) \mid d(r(t), R(t)) < \varepsilon\}$, d is a selected distance.

Remark 4.2.1. *Definition 4.2.3 generally defines the reliable degree of a data rule without any specification of the distance measurement d and acceptable error scale ε. It leaves the freedom to user for selecting the appropriate definitions in a domain-specific situation. Moreover, Definition 4.2.3 is a kind of performance measurement of a data rule. Since this measurement is only determined by a given observing period and an acceptable error scale, data rules with different forms are able to be compared under the same standard.*

By Definition 4.2.3, the following proposition holds.

Proposition 4.2.2. *Given an observing period* T *and two acceptable error scales* ε_1, ε_2, *if* $\varepsilon_1 \leqslant \varepsilon_2$, *then* $\tau(r, \varepsilon_1) \leqslant \tau(r, \varepsilon_2)$.

The reliable degree and the feasible degree of a data rule have close relationship. As shown in Figure 4.1, the observing period $T = \{t | t = 0, 1, 2, \ldots, 11\}$ and the data rule generates six estimations at $t = 1, 3, 6, 8, 9, 11$, respectively. Hence, the feasible degree of the data rule is 0.5; while its reliable degree is 0.83 because five out of six estimations are acceptable. Moreover, it is noticed that

$$\alpha(r, T, \varepsilon) \triangleq \delta(r, T) \cdot \tau(r, \varepsilon) = \frac{|T(r)|}{|T|} \cdot \frac{|T(r, \varepsilon)|}{|T(r)|} = \frac{|T(r, \varepsilon)|}{|T|} \tag{4.3}$$

is a value determined by data rule r, observing period T, and acceptable error scale ε. Generally speaking, we can find lots of data rules which are expressed in different forms and all meet the given $\alpha(r, T, \varepsilon)$, T, and ε. Hence, users have opportunity to select the most appropriate data rule in a specific situation.

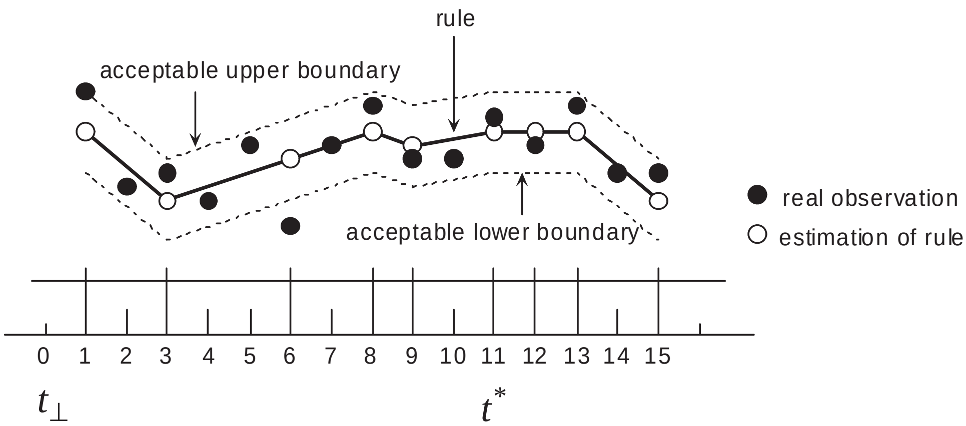


Figure 4.1: The feasible degree and reliable degree of a data rule

4.2.2.3 Covering relationship between data rules

To further compare the performances of different data rules, a covering relationship is defined based on the feasible and reliable degrees. Then a rule-map is established based on the covering relationship.

Definition 4.2.4. *Two data rules r_1 and r_2 are called equivalent to each other and are denoted by $r_1 \equiv_T r_2$, if $T(r_1) = T(r_2)$ and $r_1(t) = r_2(t)$ for any $t \in T^*$, where $T^* = T(r_1) = T(r_2)$ is an observing period.*

The equivalence relationship indicates that both data rules have the same performance, i.e., two data rules generate same estimation at the same time slot no matter what forms they are with. The equivalence relationship commonly exists in multiple descriptive models. For instance, a mathematician may use the data rule r_3 in Example 4.2.1 to describe the Fibonacci sequence; a primary school student can use the data rule r_1 to describe the same sequence although he or she does not know the sequence's exact name. Obviously, both rules have the same performance for $t > 2$.

Based on the equivalence relationship between data rules, data rules can be grouped into different clusters. Data rules in the same cluster have same performance and can be selected and used freely in an application.

Definition 4.2.5. *Data rule r_1 is called covering data rule r_2 in a given observing period T, denoted by $r_1 \succ r_2$ (or $r_2 \prec r_1$), if $T \supseteq T(r_1) \supset T(r_2)$ and $r_1(t) = r_2(t)$ for any $t \in T(r_2)$.*

Definition 4.2.5 indicates that if rule r_2 is covered by rule r_1 then r_1 has better performance than r_2 because r_1 produces more acceptable estimations than r_2. Under the covering relationship, rule r_2 is treated as a redundant data rule with respect to rule r_1.

A feature of the covering relationship is that it may turn into an equivalent relationship sometime by adjusting the observing period T (see Example 4.2.1). Because an observing period can be seen as a time window in a real application, this feature indicates that users can select applied rules through adjust time window.

Example 4.2.1. *Suppose $t_\perp = 1$ and $O(1) = 1$, $O(2) = 1$, let us consider the three data rules for the Fibonacci sequence:*

$$r_1 : r(t) = O(t-1) + O(t-2),\ (t > 2),$$

$$r_2 : r(t) = \left\lfloor \frac{O(t-1)(1+\sqrt{5})+1}{2} \right\rfloor, (t > 1), \textit{where } \lfloor x \rfloor \textit{ is the floor function, and}$$

$$r_3 : r(t) = \frac{(1+\sqrt{5})^t - (1-\sqrt{5})^t}{2^t \cdot \sqrt{5}}, (t \geqslant 1).$$

Obviously, $r_3 \succ r_2$ and $r_2 \succ r_1$ when $T = \{1, 2, \ldots\}$, although all three rules actually describe the same sequence. However, if we set $t_\perp > 2$, then the three rules are equivalent to each other.

The following conclusion for the covering relationship holds.

Proposition 4.2.3. *The covering relationship defined in Definition 4.2.5 is an equivalence relationship on the set of data rules, i.e.*

(1) for any r, $r \succ r$;

(2) for any rules r_1, r_2, and r_3, if $r_1 \succ r_2$ and $r_2 \succ r_3$, then $r_1 \succ r_3$;

(3) for any rules r_1 and r_2, if $r_1 \succ r_2$ and $r_2 \succ r_1$, then $r_1 \equiv_T r_2$. ■

4.2.2.4 Rule-map of data rules

By Proposition 4.2.3, all data rules can be grouped into several equivalent classes. In each equivalent class, we can select one data rule as a representative. On the selected representative set, a rule-map is established.

The establishment of a rule-map requires several steps. First, selected data rules are marked as nodes in a graph and are connected by directed arcs which represent the covering relationships among them. If data rule r_1 covers data rule r_2, then r_1 and r_2 are the starting and ending vertexes of the directed arc, respectively. Secondly, the directed graph is adjusted following Principle 4.2.1, 4.2.2 and 4.2.3. Finally, a hierarchical directed graph, as shown in Figure 4.2, is obtained which will be called a rule-map (denoted by $G_{\mathcal{T}}$) of data rules.

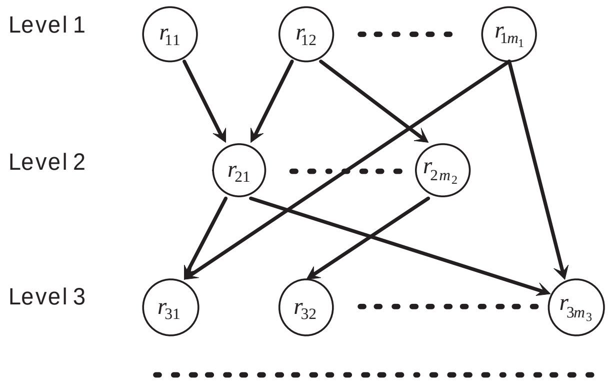


Figure 4.2: A rule-map of data rules

The following principles are used to adjust the directed arcs in a rule-map.

Principle 4.2.1. *For any rules r_1, r_2, and r_3, if $r_1 \succ r_2$ and $r_2 \succ r_3$, then the edge connecting r_1 and r_3 will be deleted from a rule-map.*

Principle 4.2.2. *For any rule r, if no rule covers r, then r will be placed at the top level.*

Principle 4.2.3. *For any rule r, r will be placed at the level as high as possible.*

By the definition of the covering relationship, three features of a rule-map are concluded as below.

(1) Data rules at the same level are incomparable to each other;

(2) Data rules at a lower level are redundant with respect to data rules at higher levels; and

(3) Data rules at a higher level have better performance than those at lower levels.

These three features are important in real applications. The first feature indicates that a functional pattern has many descriptive models in different forms. Therefore, it is possible to use a rule-map to learn and simulate the pattern in various ways and, then, get a comprehensive image of the pattern by combining the merit of each descriptive model. The second and the third features indicate that we are able to use a small number of data rules to learn and simulate the functional pattern without losing of performance. Hence, these three features guarantee the application of the rule-map technique in a complex decision environment.

4.2.2.5 Rule-map adjustment

Seen from its definition, a rule-map is closely related to the observing period T. When the observing period T varies, the structure of a rule-map must be modified accordingly. The modification mainly includes adding or removing some data rules and adjusting the positions of existed data rules in a rule-map. Rule-map adjustment provides a mechanism to trace the change of a situation.

To add and remove data rules from a rule-map, two corresponding algorithms are proposed and used below.

Algorithm 4.2.1 (Adding a data rule to a rule-map)**.** *To add a data rule r to a rule-map G_T, the following steps are conducted: set $R_r^{\succ} = \{\tilde{r} \in G_T \mid \tilde{r} \succ r\}$ and $R_r^{\prec} = \{\tilde{r} \in G_T \mid \tilde{r} \prec r\}$,*

- *For any rule $\tilde{r} \in R_r^{\succ} \cup R_r^{\prec}$, connect $\tilde{r}$ and r by a directed arc.*
- *Remove arcs connecting rules between $R_r^{\succ}$ and $R_r^{\prec}$.*
- *Adjust G_T^* by Principles 4.2.1, 4.2.2 and 4.2.3.*

Then a new rule-map G_T^* is obtained.

Algorithm 4.2.2 (Removing an existing data rule from a rule-map)**.** *To remove an existing data rule r from a rule-map G_T, the following steps are operated: set $R_r^{\succ} = \{\tilde{r} \in G_T \mid \tilde{r} \succ r\}$ and $R_r^{\prec} = \{\tilde{r} \in G_T \mid \tilde{r} \prec r\}$,*

- *Connect rules between $R_r^{\succ}$ and $R_r^{\prec}$.*
- *Remove arcs connecting rule $\tilde{r} \in R_r^{\succ} \cup R_r^{\prec}$ and rule r. Then remove rule r from G_T.*
- *Rearrange G_T^* by Principle 4.2.1, 4.2.2 and 4.2.3.*

Then a new rule-map G_T^* is obtained.

To adjust data rules' position, we apply Principle 4.2.1, 4.2.2, and 4.2.3 repeatedly. Figure 4.3 gives an example of adjusting a rule-map.

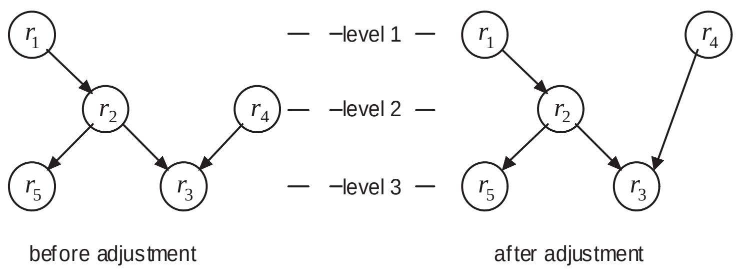


Figure 4.3: An example of adjusting a rule-map

4.2.3 A rule-map-based data inconsistency detection method

Based on the rule-map technique established in Section 4.2.2, this section presents a method for the data inconsistencies detection problem illustrated in Problem 4.2.1. This method is called a rule-map-based data inconsistency detection (RMDID) method. The framework and steps of the RMDID method are shown in Figure 4.4.

The RMDID method includes five main steps. Details of each step are illustrated below.

Step 1: Establish data rules from historical observations.

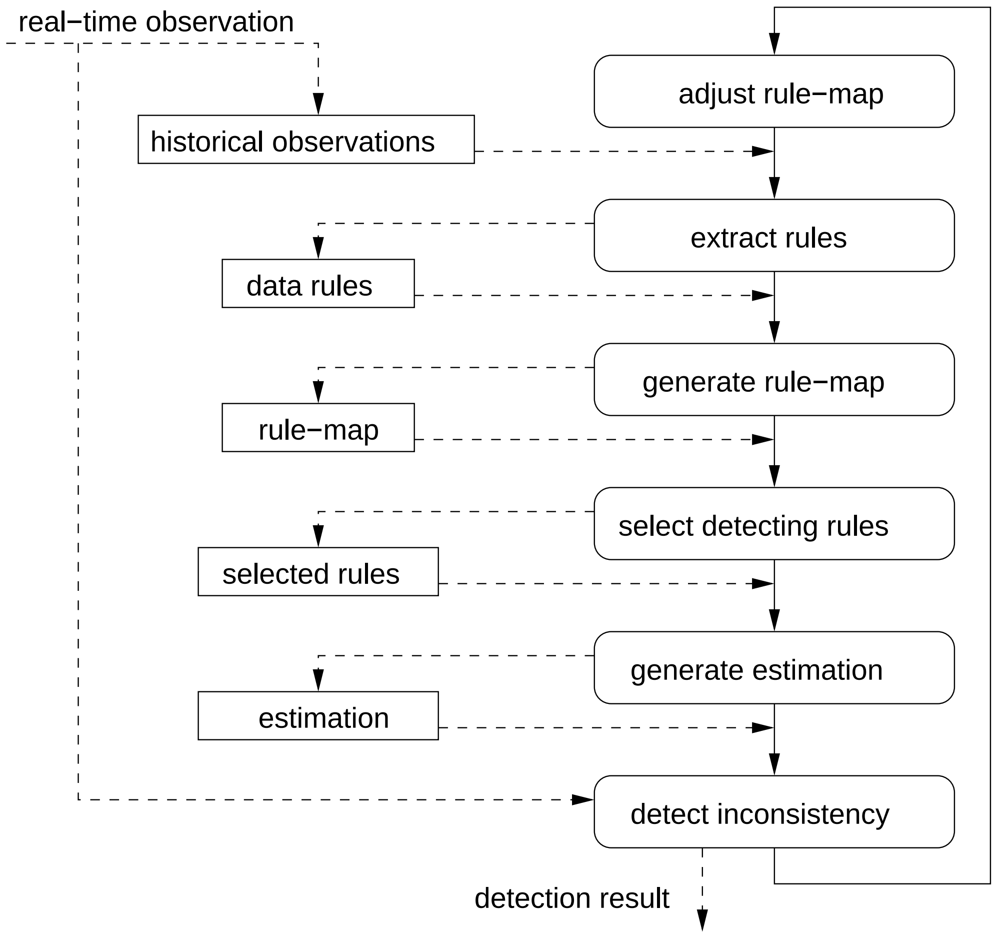


Figure 4.4: Main steps of the RMDID method

To establish data rules from H, we can use methods reported in research fields such as data mining, statistics, machine learning. The established data rules will be expressed in the form defined in Definition 4.2.1 and their feasible degree and reliable degree will be calculated accordingly.

Step 2: Construct a rule-map based on established data rules.

According to the covering relationship defined in Definition 4.2.5 and Principle 4.2.1, 4.2.2, and 4.2.3, a rule-map G can be established.

Step 3: Select data rules from the established rule-map.

By the three features of a rule-map (described in Section 4.2.2.2), we first select rules from the top level of G as candidate detection rules. Secondly, considering that some data rules may not work at a given time slot t, we remove those fail-to-work rules from the candidate detection rules. Considering an example as shown in Figure 4.5, suppose rules r_1, r_2, r_3, and r_4 are four candidate detection rules which are at the top level of a rule-map. At time slot 1, the selected detection rules are r_1, r_2 and r_4 because rule r_3 does not produce estimation. At time slot 2, the selected detection rules are r_1, r_2, and r_3 for same reason. This example indicates that selecting detection rules is a dynamic processing.

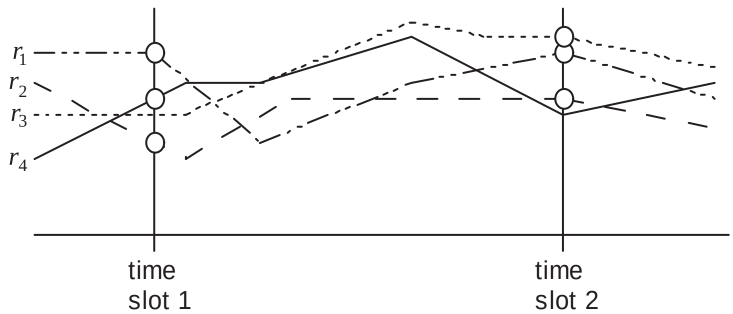


Figure 4.5: Selecting detection rules at time slot 1 and time slot 2

Step 4: Generate a reference standard using selected detection rules.

Let the selected detection rules be r_1, r_2, ..., r_n. Each data rule will present an estimation of the reference standard. Suppose these estimations are $r_1(t)$, ..., $r_n(t)$, then the generated estimation $\tilde{R}(t)$ of the reference standard for detecting inconsistency of observation $O(t)$ is given by

$$\tilde{R}(t) = \mathrm{Agg}(r_1(t), r_2(t), \ldots, r_n(t)), \tag{4.4}$$

where Agg is a selected aggregation operator (Beliakov and Warren, 2001; Yager, 2005). Here, the selection of an aggregation operator is determined by the nature of the observations and the applied data rules. For example, if these estimations are expressed in numbers, we can select the generalized weighted-sum as:

$$\tilde{R}(t) = \frac{\sum_{i=1}^{n} \tau_i r_i(t)}{\sum_{i=1}^{n} \tau_i}, \tag{4.5}$$

where τ_i is a parameter related to data rule r_i (for example, τ_i may be the reliable degree of data rule r_i). The generated estimation will be used as the reference standard for detecting inconsistency in a real-time observation.

Step 5: Detect data inconsistency in a real-time observation.

Let ε be an acceptable error scale and $d(x, y)$ be a given distance measurement. If $d(O(t), \tilde{R}(t)) < \varepsilon$, then the observation $O(t)$ is treated as an acceptable observation, i.e. it is consistent with the functional pattern. Otherwise, the observation is inconsistent with the function pattern.

Through the five steps illustrated above, potential data inconsistency in a real-time observation can be detected.

Remark 4.2.2. *In Step 4, the RMDID method uses a generated estimation as the reference standard. The support for this process is the monotonous property of aggregation operators* Agg *(Mesiar et al., 2008a):*

$$\max\{a_1, a_2, \ldots, a_n\} \geqslant \mathrm{Agg}(a_1, a_2, \ldots, a_n) \geqslant \min\{a_1, a_2, \ldots, a_n\}. \tag{4.6}$$

This property means that the generated estimation is also acceptable provided that the estimation from each single descriptive model is acceptable. In other words, that taking the generated estimation as the reference standard is a trade-off of estimations of multiple descriptive models.

4.2.4 Experiments and analysis

In order to test the proposed RMDID method, we select two data sets from Hyndman's "Time Series Data Library" (Hyndman, 2006) as testing cases and take some data fitting models in the CurveExpert software (Hyams, 2007) as established data rules to conduct the experiments.

The first experiment is based on the data set "bicup2006.dat" which records the number of passenger arrivals in a subway bus terminal in Santiago de Chile for each 15-minute interval between 6:30AM and 22:00PM from 1 March 2005 to 21 March 2005. The reason for selecting this data set is that passengers' arrival follows an internal pattern which varies from working days to weekends and from peak hours to off-peak hours. This internal pattern can be modelled from various perspectives by using different methods such as stochastic processes. Thus we have many distinctive models of the pattern and can use them to test the proposed rule-map technique in a dynamic situation.

The second experiment is based on the data set "CO2.dat" which records the percentage of carbon dioxide output from gas furnace. The reason for selecting this data set is that industrial applications always have rigorous requirements on the stability of their equipment functions and demand high-reliability descriptive models. Hence, this data set can be used to test the fitness of the proposed method in a situation with rigorous requirements.

The main steps and results of these two experiments are described and discussed as follows.

4.2.4.1 Experiment on passenger arrival data set

The data set (bicup2006.dat) has 1323 entries distributed evenly during 21 days, i.e. 63 entries per day. We select the first 882 entries (for 14 days including 10 working days and 4 weekends) as training data to construct a rule-map and the entries for 20

March 2005 (weekend) and 21 March 2005 (working day) as testing data.

For convenience, each day is marked as d_i, $i = 1, \ldots, 21$. Let $d_i(t)$ represent the t-th entry of day d_i, where $t = 1, \ldots, 63$.

Step 1: Establish data rules. We apply three methods to establish data rules from the training data.

The first method is to divide the training data into two parts, one for working days (WD) and the other for weekends (HD), and establish a data rule from each part, respectively. Here, for simplicity, we use the arithmetic mean to establish data rules for working days and weekends. For each time slot t, let

$$r_{WD}(t) = \frac{\sum_{d_i \in WD} d_i(t)}{|WD|}, \quad t = 1, 2, \ldots, 63, \tag{4.7}$$

and

$$r_{HD}(t) = \frac{\sum_{d_i \in HD} d_i(t)}{|HD|}, \quad t = 1, 2, \ldots, 63, \tag{4.8}$$

where $WD = \{d_1, d_2, d_3, d_4, d_7, d_8, d_9, d_{10}, d_{11}, d_{14}\}$, $HD = \{d_5, d_6, d_{12}, d_{13}\}$. These two data rules indicate the estimations for working days and weekends are constants for any time slot t in testing days.

The second method first computes the 14-day mean for each time slot t and uses the obtained 63 mean values to build a new data set. For the new data set, the 63 entries are divided into four parts corresponding to four periods, i.e. early morning, morning, afternoon and evening periods (shown in Table 4.1). Secondly, on each part, a statistical model is established to fit entries in that part. Therefore, we have four descriptive models for the four periods, respectively. The four statistical models used are shown in Table 4.2.

The third method is taken from Das *et al.* (1998) and applied on the 14-day-mean data set obtained in the second method. For convenience, we make the data in the 14-day-mean data set as $s_1, \cdots, s_{63}$. Then setting the clustering distance to $d = 20$

Table 4.1: Period participation for 63 time slots

periods	recording time slots	time slot indexes (t)
Early Morning	6:30AM - 8:00AM	1-7
Morning	8:15AM - 12:00 AM	8-23
Afternoon	12:15AM - 18:00 AM	24-47
Evening	18:15AM - 22:00 AM	48-63

Table 4.2: Rules for each period

periods	rules	parameters
Early Morning	$r_{EM}(t) = 1/(a + bt + ct^2)$	$a = 0.760$, $b = -0.181$, $c = 0.012$
Morning	$r_M(t) = 1/(at + b)$	$a = -0.004$, $b = 0.115$,
Afternoon	$r_A(t) = a + bt + ct^2 + dt^3$	$a = -521.284$, $b = 51.813$, $c = -1.573$, $d = 0.015$
Evening	$r_E(t) = a + bt + ct^2$	$a = 11.537$, $b = 4.891$, $c = -0.080$

(which means the average error of passenger arrivals is at most 9 (person) at each time slot t) and the slide window to 5, we generate 9 clusters; and each cluster represents a period covering 5 sequential time slots. The 9 clusters are marked by 9 symbols $\{a_i \mid i = 1, \ldots, 9\}$. Next, we translate the data set $s_1, \cdots, s_{63}$ to another data S which contains 59 symbols in $\{a_i \mid i = 1, \ldots, 9\}$. On S, we establish a rule

$$r_S : s(j) = s(j-1), \quad j = 2, \ldots, 59 \tag{4.9}$$

where $s(j)$ is j-th symbol in S.

By this rule and the L_2 distance metric, we obtain an interval of possible estimations at time slot t ($t \geqslant 6$). The interval is calculated as follows.

Let $s(j-1)$ and $s(j)$ represent sequences "b_0, b_1, b_2, b_3, b_4" and "b_1, b_2, b_3, b_4, b_5" respectively, where $b_i \in \{s_1, \ldots, s_{63}\}$. If $s(j-1)$ belongs to cluster p, then $s(j)$ belongs to cluster p by the data rule in Eq. (4.9). Suppose the centre of cluster p is

$(p_1, p_2, p_3, p_4, p_5)$ ($p_k \in \{s_t\}$, $k = 1, \ldots, 5$), the solution of

$$\sqrt{(b_1 - p_1)^2 + (b_2 - p_2)^2 + (b_3 - p_3)^2 + (b_4 - p_4)^2 + (x - p_5)^2} \leqslant d, \qquad (4.10)$$

gives the range of b_5 such that $s(j)$ falls in cluster p.

Because the solution of Eq. (4.10) is an interval, we take the middle point of it as the estimation produced by data rule r_S.

To calculate the reliability of above-obtained rules, we set the acceptable error scale to 5 (person) which means if the difference between an estimation and a real record is less than or equal to 5, then the estimation is treated as correct. Table 4.3 compares the extracted rules for experiment 1.

Table 4.3: Extracted rules of the passenger arrival example

Rules	Days applied to	Time slots applied to	Feasibility	Reliability
r_{WD}	Working days	$1 \leqslant t \leqslant 63$	71.43%	61.59%
r_{HD}	Weekends	$1 \leqslant t \leqslant 63$	28.57%	96.43%
r_{EM}	All	$1 \leqslant t \leqslant 7$	11.10%	79.59%
r_M	All	$8 \leqslant t \leqslant 23$	25.40%	33.48%
r_A	All	$24 \leqslant t \leqslant 47$	38.10%	19.05%
r_E	All	$48 \leqslant t \leqslant 63$	25.40%	25.45%
r_S	All	$6 \leqslant t \leqslant 63$	92.06%	16.01%

Step 2: Construct a rule-map. By comparing the rules in Table 4.3, we note that they are incomparable to each other. For example, rule r_{WD} and rule r_{HD} cover different days; and rule r_{EM} and r_M cover different time slots. Hence, all rules are at the top level of a rule-map G for this experiment.

Step 3: Select detection rules. For the testing data set, the selected rules are listed in Table 4.4.

Step 4: Generate reference standard. By the selected rules, the estimations for different time slots can be computed. For instance, to get the estimation of day $d_{20}(20)$, we use the rules r_{HD}, r_M, r_S and obtain three estimations $r_{HD}(20) = 1$, $r_M(20) = 22$, $r_S(20) = 15$. Hence, the estimation is 8 by Eq. (4.5).

Table 4.4: Selected rules for day 20 and day 21

Day (i)	Time slots (t)	Selected rules	Day (i)	Time slots (t)	Selected rules
20	$\{1,\ldots,5\}$	r_{HD}, r_{EM}	21	$\{1,\ldots,5\}$	r_{WD}, r_{EM}
	$\{6,7\}$	r_{HD}, r_{EM}, r_S		$\{6,7\}$	r_{WD}, r_{EM}, r_S
	$\{8,\ldots,23\}$	r_{HD}, r_M, r_S		$\{8,\ldots,23\}$	r_{WD}, r_M, r_S
	$\{24,\ldots,47\}$	r_{HD}, r_A, r_S		$\{24,\ldots,47\}$	r_{WD}, r_A, r_S
	$\{48,\ldots,63\}$	r_{HD}, r_E, r_S		$\{48,\ldots,63\}$	r_{WD}, r_E, r_S

Step 5: Detect data inconsistency. The obtained passenger arrivals of $d_{20}(20)$ is 0. Obviously, the observation is not acceptable to the expected estimation if the acceptable error scale is set to 8 or less. Hence, it is inconsistent with the main part of the pattern.

Now, considering the case of $d_{21}(20)$, we select the rules r_{WD}, r_M, r_S and obtain the estimations 27 by Eq. (4.5). The real observation for $d_{21}(20)$ is 35. Therefore the real observation is consistent if the acceptable error is set to 8.

4.2.4.2 Experiment on carbon dioxide data set

The data set "CO2.dat" contains 296 entries. We use this data set mainly to test the performance of the proposed method under different acceptable error scales.

In this experiment, the first 100 entries are selected as training data and the followed 20 entries as testing data. Then four acceptable error scales are defined, i.e. 0.005, 0.01, 0.03, and 0.05, which mean the deviation between expected observation (i.e., the reference standard) and reported observation is equal to or less than 0.005, 0.01, 0.03, and 0.05, respectively. The applied data rules are selected from the CurveExpert software (Hyams, 2007) and shown in Table 4.6. Table 4.7 displays their feasible and reliable degrees. According to the definition of the covering relationship, these data rules are incomparable to each other; and they are all located at the top level of a rule-map and are used as selected rules.

The overall result of the experiment is summarised in Table 4.5. In Figure 4.6 the

Table 4.5: Testing result of carbon dioxide example

Acceptable error scales	0.005	0.01	0.03	0.05
Number of inconsistent entries in the training set	82	69	33	5
Number of inconsistent entries in the testing set	16	11	5	4

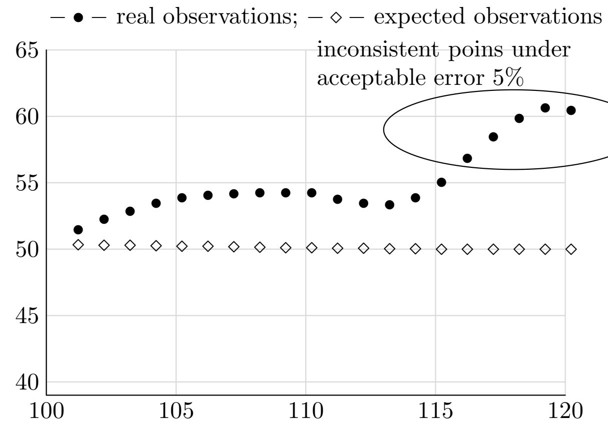


Figure 4.6: Inconsistent points under acceptable error scale 0.05

circled elements are the inconsistent points under the acceptable error scale 0.05. In Figure 4.7, the circled elements are the inconsistent points under the acceptable error scale 0.03. Figure 4.8 compares the reference standard generated by a rule-map, the estimations generated by the selected rules, and the real observations.

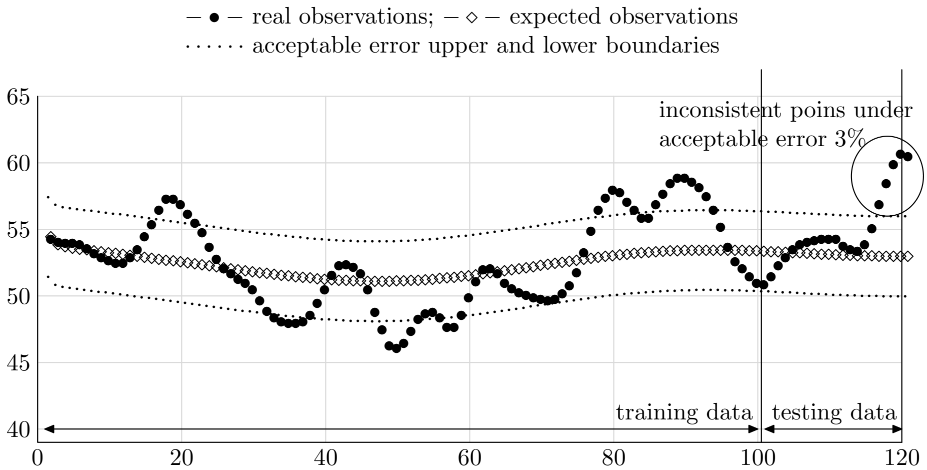


Figure 4.7: Inconsistent point under acceptable error scale 0.03

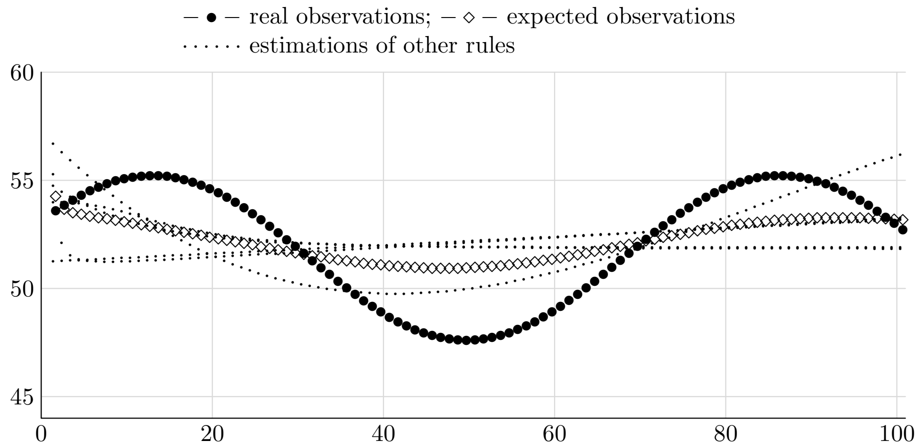


Figure 4.8: Comparison of real observations, expected observations (reference standard), and estimations by description models

Table 4.6: Statistical models used for the carbon dioxide experiment

Model expression	Model parameters
$r_1(t) = a + bt + c/t^2$	$a = 50.71$, $b = 0.02$, $c = 4.26$
$r_2(t) = (ab + ct^d)/(b + t^d)$	$a = 51.55$, $b = 0.00064$, $c = 53.72$, $d = -2.72$
$r_3(t) = a + bt + ct^2 + dt^3$	$a = 56.80$, $b = -0.40$, $c = 0.0064$, $d \approx 0.0$
$r_4(t) = a/(1 + be^{-ct})$	$a = 51.62$, $b = -0.0572$, $c = 0.092$
$r_5(t) = a + b\cos(ct + d)$	$a = 51.14$, $b = 3.81$, $c = 0.0856$,$d = -1.05$
$r_6(t) = a/(1 + e^{(b-ct)^{1/d}})$	$a = 59.26$, $b = 22.86$, $c = 0.05698$, $d = 151.45$

4.2.4.3 Experimental results analysis

Due to the fact that real observations are presented with inevitable errors, experiments indicate that the proposed method can appropriately describe the trend of a functional pattern in the sense that both the produced estimations and the real observations have similar fluctuation. In particular, we do not use any prior knowledge in these experiments. The majority of estimations of a rule-map fall in the acceptable boundaries although they do not have the same values as real observations (such as shown in Figure 4.6). This fact means our method indeed grasps the main functional pattern of the monitored object. Moreover, experiments indicate the proposed method can provide suggestions about anomalies to some extent. For example, the circled points shown in Figure 4.6 are observations whose values exceeds the given acceptable error boundaries. These observations can be treated as inconsistent points and

Table 4.7: Data rules' performance measurements (Unit: percents)

Data rules	FD	RD (0.05)	RD (0.03)	RD (0.01)	RD (0.005)
$r_1(t) = a + bt + c/t^2$	100	48	28	10	2
$r_2(t) = (ab + ct^d)/(b + t^d)$	100	53	39	18	12
$r_3(t) = a + bt + ct^2 + dt^3$	100	32	21	8	3
$r_4(t) = a/(1 + be^{-ct})$	100	53	39	19	9
$r_5(t) = a + b\cos(ct + d)$	100	37	21	6	3
$r_6(t) = a/(1 + e^{(b-ct)^{1/d}})$	100	0	0	0	0

FD=Feasible Degree, RD=Reliable Degree

should be paid more attention to. Experiments also indicate that the proposed RMDID method can take the advantage of multiple descriptive models. For instance, although each single data rule used in Experiment 2 has relatively low reliability, the generated reference standard by a rule-map, however, is near the function pattern simulated. Finally, the proposed RMDID method is established on an open framework, which is mainly incarnated in the definition of reliable degree of a data rule and the generated estimation. In the definition of reliable degree, the distance between the estimation (the expected observation) and the reported observation is not specified, which brings facility to meet the demands of a variety of descriptive models. The aggregate operator for generating the reference standard in Eq. (4.4) can be selected following different decision preferences. Hence, the proposed method is adaptable in a dynamic decision environment.

However, we also note that the proposed RMDID method still has some drawbacks. First, the performance of the RMDID method is affected by the selected models. As shown in Section 4.2.4.1, the data rules r_{EM} and r_M produce wrong estimations for weekends because they do not distinguish weekends from working days. A possible solution to enhance the performance of the RMDID method is to improve the reliable degrees of the selected data rules. Moreover, in some cases, the RMDID method may face the risk of getting incorrect conclusion due to the wrong estimations of the selected detection data rules. To address these drawbacks, further research is still to be conducted.

4.3 Logical inconsistency detection

4.3.1 Logical inconsistency detection problem

Corresponding to data inconsistency problem, this section discusses solutions of logical inconsistency detection problem. Formally, a logical inconsistency detection problem is defined below.

Problem 4.3.1 (Logical inconsistency detection problem)**.** *In a decision environment, a set of objects $O = \{O_1, \ldots, O_n\}$ are monitored. Knowledge about these objects has been stored during a period T. At a moment t (in general, $t \notin T$), we collect some observations on the monitored objects and want to know whether these observations have potential logical inconsistency with each other and with our stored knowledge.*

In Problem 4.3.1, the monitored objects and their observations are open to define. In a real application, the monitored objects can be some measuring sensors, the indicators of an emergency event, or the indexes of a running business. Correspondingly, the collected observations of them can be the readings of measuring sensors, states of indicators for emergency events, or predicted trends of business indexes.

Without loss of generality, let T be the time period over which the knowledge of an application domain are collected. The stored knowledge over T is denoted by Ω and $\Omega = \cup_{t \in T}\Omega(t)$, where $\Omega(t)$ is the stored knowledge at t.

4.3.2 State-based domain knowledge representation

This section will present a state-based knowledge representation method.

4.3.2.1 Overview

Domain knowledge can be expressed by combinations of objects' states. Usually, a piece of domain knowledge gives an illustration of some objects and their states. Because object's state varies at different time, the combinations of their states are changeable. The changed combinations describe the relationships between these objects in the domain. The change of these objects' states and the relationships between these objects form a main part of the knowledge we have about the domain. Hence, the state combinations can be used as a model to represent the domain knowledge.

Using the combination of objects' states as representation of domain knowledge has three different meanings. Firstly, a combination can link many different objects. This means that the combination illustrates the relationship between objects. Secondly, a combination can link different states of a single object. This means that the combination illustrates the similarity between an object's states. Thirdly, two combinations at different time can illustrate the dynamic change of an application domain. Hence, the combination of monitored objects' states can be used to a dynamic decision environment.

Considering that an object's state is the fundamental element in a state combination, this section calls the developed knowledge representation method as a state-based knowledge representation method.

4.3.2.2 Domain knowledge and state combination

Suppose D is an application domain, the knowledge of D is related to a set of objects denoted by O_1, O_2, ..., O_n. Each object O_i has some possible states, namely $s_1^{(i)}$, $s_2^{(i)}$, ..., $s_{i_m}^{(i)}$ (denoted by S_i) in domain D. At a given time t ($t \in T$), the normal states of object O_i are denoted by $S_i(t)$ ($S_i(t) \subseteq S_i$), $i = 1, 2, \ldots, n$, which occur in some knowledge stored in $\Omega(t)$. In the following, let $S_i(t)$ be a non-empty set without other specification.

Remark 4.3.1. *Although an object intuitionally displays a unique characteristic state at a given time t, clearly identifying that state is very difficult sometimes. In this situation, people prefer some uncertain descriptions of the state. Such an uncertain description often covers several possible states and assigns different possibilities or probabilities to them. For simplicity, this study does not process uncertain descriptions directly and assumes that the semantics of an object's states are completely distinguishable. Under this assumption, this study does not process state descriptions like "old" or "young"; but does process state descriptions like "red, yellow, blue, white" or "A, B, C, D, E".*

Before given the definition of domain-specific knowledge, we first introduce the notion of unordered n-tuples.

Definition 4.3.1. *Let X_1, ..., X_n be n non-empty sets and $X_i \neq X_j$ if $i \neq j$ for any $i, j = 1, \ldots, n$. Suppose $x_i \in X_i$, $i = 1, \ldots, n$, then an unordered n-tuple of x_1, x_2, ..., x_n is denoted by $(x_1, x_2, \ldots, x_n)$ where the order of x_1, x_2, ..., x_n is not considered.*

In the following, the set of all unordered n-tuples generated from X_1, X_2, ..., X_n is denoted by $X_1 \otimes X_2 \otimes \cdots \otimes X_n$.

Based on Definition 4.3.1, we can define a representation for knowledge of a domain. Our idea is simple: at a given time t, let $J(t)$ be the set of indexes of objects whose states are collected at t and X_i be the state set $S_j(t)$ of object O_j at time t, $j \in J$; then a piece of knowledge of the application domain is a subset of $\otimes_{j \in J} S_j(t)$. The detailed definition is given below.

Definition 4.3.2. *A piece of domain-specific knowledge $\omega(t)$ of domain D at time t is a set of unordered p-tuples, i.e.,*

$$\omega(t) \subseteq \bigotimes_{j \in J(\omega(t))} S_j(t), \tag{4.11}$$

where $J(\omega(t)) \subseteq \{1, 2, \ldots, n\}$ *is the index set of objects described by the knowledge* $\omega(t)$*, and* $p = |J(\omega(t))|$.

In the following, we use the term "knowledge" for "domain-specific knowledge" without other specification and use $S_j^{(\omega(t))}$ to denote the states of object O_i described by the knowledge $\omega(t)$.

A piece of knowledge $\omega(t)$ is known as empty knowledge if for some $j \in J(\omega(t))$, $S_j^{(\omega(t))}$ is an empty set. Empty knowledge indicates a state combination is non-existent. Moreover, empty knowledge may have different forms with the differences of time t and index $j(\in J)$.

Definition 4.3.2 can be used to explain some commonly used inference rules between two objects. There are three typical inference relationships between two objects A and B, i.e., $A \Rightarrow B$ (i.e., A implies B), $B \Rightarrow A$ (i.e., B implies A), and $A \Leftrightarrow B$ (i.e., A implies B and vice verse). When $A \Rightarrow B$, this means from a given state of A, some states of B can be obtained. We can use a piece of knowledge, which is composed of unordered 2-tuples, to describe this relationship as follows

$$\omega_{A \Rightarrow B} = \{(a, b) | b \in S_B\}. \tag{4.12}$$

Similarly, relationship $B \Rightarrow A$ can be expressed by a piece of knowledge

$$\omega_{B \Rightarrow A} = \{(a, b) | a \in S_A\}. \tag{4.13}$$

Because relation $A \Leftrightarrow B$ often indicates the corresponding between particular states of A and B, the knowledge for describing this relationship is of form

$$\omega_{A \Leftrightarrow B} = \{(a_i, b_i) | a_i \in S_A, b_i \in S_B, i = 1, \ldots, q\}. \tag{4.14}$$

.

Collecting all domain knowledge over a time period, we can establish a domain knowledge base.

Definition 4.3.3. *The knowledge base (Ω) of domain D before time t_c is a non-empty set of domain-specific knowledge $\omega(t)$ such that*

$$\Omega = \bigcup_{t<t_c, t\in T} \Omega(t), \tag{4.15}$$

and $\Omega(t)$ is the knowledge based at time t.

Definition 4.3.3 indicates that the domain knowledge base is composed of a set of state combinations of monitored objects. Moreover, the knowledge base may be incomplete because it consists of knowledge known prior to a particular time. This feature is in accordance with people's cognitive experience. Moreover, the domain knowledge base may include duplicate knowledge because some knowledge is correct in multiple time slots. Finally, the knowledge base may include both consistent and inconsistent knowledge because some knowledge is only applicable to some special circumstances.

By Definitions 4.3.2 and 4.3.3, we can identify three possible relationships between a state combination c and a knowledge base Ω, i.e.,

- c is an existed combination in Ω. In this case, c occurs in some pieces of knowledge in Ω. In other words, for some t, there exists at least a piece of knowledge $\omega_c \in \Omega(t)$ such that $c \in \omega_c$.

- c is a potential combination in Ω. In this case, c is currently not in any piece of knowledge in Ω; but it is a possible state combination, i.e., each state in it is a normal state.

- c is a non-existent combination. In this case, c does not occur in any piece of knowledge in Ω and at least one of its states is not a normal state.

The three relationships between a state combination and a domain knowledge base can help us to make a strategy to solve Problem 4.3.1. Because a real observation can be expressed by a state combination of related objects, we can a real observation's logical inconsistency by considering three cases. The first case means that the obtained state combination has been recognized and stored. There is no need to check the consistency of the obtained state combination provided that the knowledge base is consistent. The second case shows that the obtained state combination has not been observed and stored previously but it may exist in the future. Hence, there is a need to check the consistency of the obtained state combination against the domain-specific knowledge base. Once it is consistent with the knowledge base, the obtained state combination can be added to the knowledge base as a piece of new knowledge. As for the third case, the obtained state combination is bound to be inconsistent with the knowledge base. Therefore, there is no need to check the consistency of it.

An important step to implement above processing strategy is to establish a consistent knowledge base. To do so, we will define the consistency of knowledge first.

4.3.2.3 State-based knowledge consistency

State-based knowledge consistency has two meanings. On the first level, it means the consistency in a knowledge base at each given time, i.e., no paradox can be derived from each $\Omega(t)$. On the second level, it means that the whole knowledge base is consistent, i.e., no contradiction can be deduced from Ω. Existing information detection methods mainly focus on the consistency of the whole knowledge base and the consistency of knowledge at each time is presumed implicitly. Because those techniques do not consider the influence of time change, the knowledge set at each time is the same. Obviously, such a knowledge base is smaller than the one given in Definition 4.3.3. Moreover, a smaller knowledge base may exclude some consistent information by mistake.

This section mainly focuses on the consistency of a real observation against a knowledge base $\Omega(t)$ at a given time t. Hence, we suppose $\Omega(t)$ is consistent.

Definition 4.3.4. *Let $\omega(t)$ be a piece of knowledge and $J = \{i_1, \ldots, i_q\}$ be a non-empty subset of $J(\omega(t))$. Then the J-trunk of $\omega(t)$ is denoted by $\omega(t)|_J$ such that:*

$$\omega(t)|_J = \{(s_{k_1}^{(i_1)}, \ldots, s_{k_q}^{(i_q)}) | \exists (s_{k_1}^{(i_1)}, \ldots, s_{k_q}^{(i_q)}, s_{k_{l_1}}^{(l_1)}, \ldots, s_{k_{l_p}}^{(l_p)}) \in \omega(t)\}.$$

Definition 4.3.4 indicates a J-trunk of a piece of knowledge $\omega(t)$ is the joint set of J-trunk of each state combination in it. By Definition 4.3.2, a J-trunk of a piece of knowledge $\omega(t)$ is also a piece of knowledge. Noted that this knowledge is obtained from $\omega(t)$, therefore, it can be seen as a logical consequence of $\omega(t)$. Keeping this feature in mind, we can define the consistency between two pieces of knowledge.

Suppose $\omega(t)$ and $\phi(t)$ are two pieces of knowledge, and J is the intersection of $J(\omega(t))$ and $J(\phi(t))$. We need to consider two situations. First, J is not an empty set. Then $\omega(t)|_J$ and $\phi(t)|_J$ have three possible relationships, i.e.,

(1) $\omega(t)|_J = \phi(t)|_J$. This relationship indicates that same conclusion is derived from both two pieces of knowledge; therefore, these two pieces of knowledge are consistent.

(2) $\omega(t)|_J \cap \phi(t)|_J \neq \emptyset$ but $\omega(t)|_J \neq \phi(t)|_J$. This relationship indicates that the logical consequences of the two pieces of knowledge have some common state combinations; therefore, these two pieces of knowledge are partly consistent although the consequences from them may not completely identical.

(3) $\omega(t)|_J \cap \phi(t)|_J = \emptyset$. This relationship indicates that the logical consequences of two different pieces of knowledge have not common state combinations; therefore, these two pieces of knowledge are inconsistent to each other.

Secondly, J is an empty set. In this situation, we cannot judge the consistency between $\omega(t)$ and $\phi(t)$ directly; but we can do that by finding some intermediate knowledge which has possible state combinations linking $\omega(t)$ and $\phi(t)$. Suppose $\psi(t)$ is a piece of knowledge and both $J(\omega(t)) \cap J(\psi(t))$ and $J(\phi(t)) \cap J(\psi(t))$ are not empty sets. We hope to find a state combination c

$$(s_1^{\omega(t)}, \cdots, s_{i_1}^{\omega(t)}, \cdots, s_{i_m}^{\omega(t)}, s_{i_m+1}^{\psi(t)}, \cdots, s_{j_1-1}^{\psi(t)}, s_{j_1}^{\psi(t)}, \cdots, s_{j_m}^{\phi(t)}, \cdots, s_{j_n}^{\phi(t)}) \tag{4.16}$$

such that

$$\begin{aligned} &(s_1^{\omega(t)}, \cdots, s_{i_1}^{\omega(t)}, \cdots, s_{i_m}^{\omega(t)}) \in \omega(t) \\ &(s_{i_1}^{\omega(t)}, \cdots, s_{i_m}^{\omega(t)}, s_{i_m+1}^{\psi(t)}, \cdots, s_{j_1-1}^{\psi(t)}, s_{j_1}^{\psi(t)}, \cdots, s_{j_m}^{\phi(t)}) \in \psi(t) \\ &(s_{j_1}^{\psi(t)}, \cdots, s_{j_m}^{\phi(t)}, \cdots, s_{j_n}^{\phi(t)}) \in \phi(t). \end{aligned}$$

If such a c exists, we can say that there is a common state combination in the logical consequences of $\omega(t)$ and $\phi(t)$ and, therefore, knowledge $\omega(t)$ and $\phi(t)$ are consistent to some extent. In the following, we call $\psi(t)$ a piece of connecting knowledge between $\omega(t)$ and $\phi(t)$ and state combination c a linking combination between $\omega(t)$ and $\phi(t)$.

Based on the above analysis, the following definitions about consistency between two pieces of knowledge are given.

For simplicity, we use J^* to denote the intersection of $J_{\omega(t)}$ and $J_{\phi(t)}$.

Definition 4.3.5. *Suppose $\omega(t)$, $\phi(t) \in \Omega(t)$ are two pieces of knowledge.*

When J^* is a non-empty set, $\omega(t)$ and $\phi(t)$ are said to be strictly consistent if $S_j^{(\omega(t))} = S_j^{(\phi(t))}$ for any $j \in J^*$; $\omega(t)$ and $\phi(t)$ are said to be partially consistent if $S_j^{(\omega(t))} \neq S_j^{(\phi(t))}$ for some $j \in J^*$ and $S_j^{(\omega(t))} \cap S_j^{(\phi(t))} \neq \emptyset$ for any $j \in J^*$; and $\omega(t)$ and $\phi(t)$ are said to be inconsistent if for some $j \in J^*$, $S_j^{(\omega(t))} \cap S_j^{(\phi(t))} = \emptyset$.

When J^* is an empty set, $\omega(t)$ and $\phi(t)$ are said to be strictly consistent if there is a piece of connecting knowledge $\psi(t)$ between $\omega(t)$ and $\phi(t)$ such that $\omega(t)$ and $\psi(t)$ are *strictly consistent and $\psi(t)$ and $\phi(t)$ are strictly consistent simultaneously; $\omega(t)$ and $\phi(t)$ are said to be partially consistent if there is a piece of connecting knowledge $\psi(t)$ between $\omega(t)$ and $\phi(t)$ such that $\omega(t)$ and $\phi(t)$ are connected by at least one linking combination but are not strictly consistently connected; $\omega(t)$ and $\phi(t)$ are said to be inconsistent if there is not linking combination between $\omega(t)$ and $\phi(t)$.*

By Definition 4.3.5, it is known that the operations of obtaining the J^*-trunk of two pieces of knowledge and finding a linking combination between them are very important for detecting the consistency. Here, we formally define two operations to conduct those two tasks. The first operation is called an extracting operation and denoted by $E(\omega, \phi)$. The second operation is called a coupling operation and denoted by $C(\omega, \phi)$.

Definition 4.3.6. *Let $\omega(t)$, $\phi(t) \in \Omega(t)$ be two pieces of knowledge and $J^* \neq \emptyset$. Then $E(\omega, \phi)$ is obtained by:*

$$E(\omega(t), \phi(t)) = \omega(t)|_{J^*} \cap \phi(t)|_{J^*}, \tag{4.17}$$

where $J^* = J(\omega(t)) \cap J(\phi(t)) \neq \emptyset$.

Definition 4.3.7. *Let $\omega(t)$, $\phi(t) \in \Omega(t)$ be two pieces of knowledge and $J^* \neq \emptyset$. Then $C(\omega, \phi)$ is obtained by:*

$$C(\omega(t), \phi(t)) = \{c | c \text{ is a linking combination between } \omega(t) \text{ and } \phi(t)\}. \tag{4.18}$$

By Definition 4.3.6 and Definition 4.3.7, the following conclusion holds.

Proposition 4.3.1. *Let $\omega(t)$ and $\phi(t)$ be two pieces of strictly (partially) consistent knowledge and $J^* \neq \emptyset$, then*

$$E(\omega(t), \phi(t)) = C(\omega(t), \phi(t))|_{J^*}. \tag{4.19}$$

Proof. For any $c \in E(\omega(t), \phi(t))$, we have $c_1 \in \omega(t)$ and $c_2 \in \phi(t)$ such that c is the common section of c_1 and c_2. Then, there is a combination $\tilde{c}$ linking $\omega(t)$ and $\phi(t)$. Hence, $\tilde{c} \in C(\omega(t), \phi(t))$. By Definition 4.3.4, we have $c|_{J^*} = c$. Obviously, $c \in C(\omega(t), \phi(t))|_{J^*}$ and $E(\omega(t), \phi(t)) \subseteq C(\omega(t), \phi(t))|_{J^*}$.

For any $\tilde{c} \in C(\omega(t), \phi(t))|_{J^*}$, there exist $c_1 \in \omega(t)$ and $c_2 \in \phi(t)$ such that $\tilde{c}$ is the common section of them. Notice that $\tilde{c} = \omega(t)|_{J^*}$ and $\tilde{c} = \phi(t)|_{J^*}$, $\tilde{c} \in E(\omega(t), \phi(t))$. Therefore, $C(\omega(t), \phi(t))|_{J^*} \subseteq E(\omega(t), \phi(t))$. □

Noted that $E(\omega(t), \phi(t))$ and $C(\omega(t), \phi(t))$ themselves are two pieces of knowledge, Definition 4.3.6 and Definition 4.3.7 give two methods of generating new knowledge from existed knowledge base. In the following, we use $\Psi(t) \models_D \psi(t)$ to confirm that knowledge $\psi(t)$ is generated from a set of knowledge $\Psi(t)$ by the two methods. Hence, $\{\omega(t), \phi(t)\} \models_D E(\omega(t), \phi(t))$ and $\{\omega(t), \phi(t)\} \models_D C(\omega(t), \phi(t))$.

Proposition 4.3.2. *Let $\omega(t)$ and $\phi(t)$ be two pieces of inconsistent knowledge, then either $E(\omega, \phi)$ or $C(\omega, \phi)$ is an empty set.*

Proof. We consider two possible situations. Firstly, suppose J^* is a non-empty set. In this case, $\omega(t)$ and $\phi(t)$ are inconsistent if for some $j \in J^*$, $S_j^{(\omega(t))} \cap S_j^{(\phi(t))} = \emptyset$. This means for any $c_1 \in \omega(t)$ and any $c_2 \in \phi(t)$, $c_1|_{J^*} \neq c_2|_{J^*}$. Hence, $c_1|_{J^*} \notin \phi(t)|_{J^*}$ and then $\omega(t)|_{J^*} \cap \phi(t)|_{J^*} = \emptyset$. Therefore, $E(\omega(t), \phi(t))$ is an empty set. By Proposition 4.3.1, $C(\omega(t), \phi(t))$ is an empty set. Secondly, suppose J^* is an empty set. In this case, there is not a combination c which links $\omega(t)$ and $\phi(t)$. Hence, $C(\omega(t), \phi(t))$ is an empty set. □

Proposition 4.3.2 and Definition 4.3.5 indicate that the consistency between two pieces of knowledge can be found by checking whether empty knowledge can be derived from them. We now extend this to a set of knowledge. In the following, $C(\omega, \phi)$ and $E(\omega, \phi)$ will be denoted by $\omega \sqcap \phi$ and $\omega \sqcup \phi$, respectively. For any $\Omega^*(t) \subseteq \Omega(t)$, let us define $C(\Omega^*(t))$ as follows

- $\Omega^*(t) \in C(\Omega^*(t))$;
- for any $\omega(t), \phi(t) \in C(\Omega^*(t)), \omega(t) \sqcap \phi(t) \in C(\Omega^*(t))$;
- for any $\omega(t), \phi(t) \in C(\Omega^*(t)), \omega(t) \sqcup \phi(t) \in C(\Omega^*(t))$.

Therefore, we can define consistency/inconsistency of a set of knowledge.

Definition 4.3.8. *A set of knowledge $\Omega^*(t)$ is called consistent if $C(\Omega_i^*(t))$ does not include empty knowledge; otherwise, it is called inconsistent.*

4.3.2.4 Knowledge covering relationship

Generally speaking, that checking a set of knowledge is consistent or not is a time-consuming task. However, we have a simplified strategy for it. To illustrate this strategy clearly, we introduce a covering relationship between two pieces of knowledge.

Definition 4.3.9. *Two pieces of knowledge $\omega(t)$ and $\phi(t)$ are said to be equivalent and denoted by $\omega(t) \equiv \phi(t)$ if $J(\omega(t)) = J(\phi(t))$ and $S_j^{(\omega(t))} = S_j^{(\phi(t))}$ for any $j \in J(\omega(t))(= J(\phi(t)))$.*

Definition 4.3.9 depicts the phenomena that a piece of knowledge can be expressed in different ways. For instance, both "2000 Sydney Olympic Game" and "the 26th Olympic Game" refer to the same Game hosted by Sydney in October, 2000.

Definition 4.3.10. *A piece of knowledge $\omega(t)$ is said to be a logical consequence of knowledge $\phi(t)$ if the following conditions hold:*

(1) $J(\omega(t)) \subseteq J(\phi(t))$,

(2) $\omega(t) = \phi(t)|_{J(\omega(t))}$.

In the following, we shall denote $\phi(t) \models \omega(t)$ if knowledge $\omega(t)$ is a logical consequence of knowledge $\phi(t)$. Obviously, two equivalent knowledge ω and ϕ are logical consequence of each other.

Following Definition 4.3.5 and Definition 4.3.10, if two pieces of knowledge $\omega(t)$ and $\phi(t)$ are strictly consistent, then the following conclusion holds.

Proposition 4.3.3. *If $\omega(t)$ and $\phi(t)$ are two pieces of strictly consistent knowledge, then*

- $\omega(t) \models E(\omega(t), \phi(t))$ *and* $\phi(t) \models E(\omega(t), \phi(t))$*; or*
- $C(\omega(t), \phi(t)) \models \omega(t)$ *and* $C(\omega(t), \phi(t)) \models \phi(t)$*.* ■

It is easy to verify that the following conclusions hold.

Proposition 4.3.4. *Let $\omega(t)$, $\phi(t) \in \Omega(t)$ and $\omega(t)$, $\phi(t) \notin \mho$*

(1) $\omega(t) \models \omega(t)$ *for any* $\omega(t) \in \Omega(t)$*.*

(2) $\omega(t) \equiv \phi(t)$ *if* $\omega(t) \models \phi(t)$ *and* $\phi(t) \models \omega(t)$*.*

(3) $\omega(t) \models \psi(t)$ *if* $\omega(t) \models \phi(t)$ *and* $\phi(t) \models \psi(t)$*.* ■

The logical consequence relationship $\models$ gives a hierarchical structure among a set of knowledge at time t. We draw the hierarchical structure in a graph according to the following principle:

$$\omega(t) \preccurlyeq \phi(t) \text{ if and only if } \phi(t) \models \omega(t), \tag{4.20}$$

where $\omega(t) \preccurlyeq \phi(t)$ means that $\phi(t)$ covers $\omega(t)$. The relationship $\preccurlyeq$ is a partial order, which is called knowledge covering relationship. The knowledge covering relationship among $C(\omega, \phi)$, $\omega(t)$, $\phi(t)$, and $E(\omega, \phi)$ is shown in Figure 4.9.

Further observing Definition 4.3.10 and Figure4.9, we know that the lengths of state combinations in $C(\Omega^*(t))$ are increasing but the numbers of state combinations in it are decreasing. By this feature, we can simplify the search of empty knowledge from $C(\Omega^*(t))$.

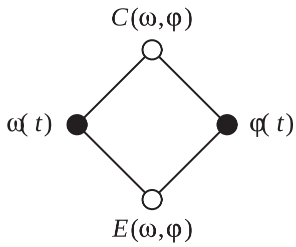


Figure 4.9: Covering among knowledge pieces

We call a set of knowledge Ω is indivisible if there doesn't exist a division of $J(\Omega) = J_1 \cup J_2 \cup \cdots \cup J_m$ such that

- $J_i \cap J_k = \emptyset$ if $i \neq k$, and
- for any $\omega \in \Omega$, there exists unique l, $J(\omega(t)) \subseteq J_l$,

where i, k, $l \in \{1, 2, \ldots, m\}$.

Proposition 4.3.5. *Suppose $\Omega^*(t)$ is a set of knowledge and $\Omega^*(t)$ is divided into m indivisible parts, $\Omega_1^*(t)$, $\cdots$, $\Omega_m^*(t)$, then*

$$C(\Omega^*(t)) = \bigcup_{i=1}^{m} C(\Omega_i^*(t)). \tag{4.21}$$

Proposition 4.3.5 indicates the empty knowledge will occurs in some $C(\Omega_i^*(t))$s. Thus, the searching space is reduced.

Remark 4.3.2. *That coupling operation on a set of knowledge aims at finding out all possible state combinations of the monitored objects. The obtained state combinations are descriptions of the internal dependencies among the monitored objects. Our knowledge of a specific domain exactly consists of this kind of descriptions. Hence, we can use these state combinations to detect inconsistency in a real observation.*

4.3.3 An information logical inconsistency detection method

Based on the state-combination-based domain knowledge representation method, this section proposes an information logical inconsistency detection method which is called state-combination-based logical inconsistency detection (SCLID) method.

To solve Problem 4.3.1, we suppose the stored knowledge $\Omega(t)$ at time t is consistent. Therefore, a potential information logical inconsistency must be introduced by the real observations S^*.

The SCLID method is composed of five steps as follows.

Step 1: Check $C(S^*)$. If $C(S^*)$ contains empty knowledge, these observations are logical inconsistent and this method stops; otherwise, the detection goes to Step 2. Step 1 aims at finding out information logical inconsistency in the observations themselves.

Step 2: Compare $J(S^*)$ and $J(C(\Omega(t)))$. If $J(C(\Omega(t))) \supseteq J(S^*)$, then the detection goes to Step 3; otherwise, the detection goes to Step 4. This step aims to determine whether the stored knowledge adapts to the needs of a logical inconsistency detection task. If the stored knowledge and these observations involve the same objects, then the stored knowledge meets the requirements of the detection task. Otherwise, some observations cannot be detected by the stored knowledge.

Step 3: Check whether $S^* \cap (C(\Omega(t)))|_{J(S^*)} \neq \emptyset$. If $S^* \cap (C(\Omega(t)))|_{J(S^*)} \neq \emptyset$, then these observations are logical consistent; otherwise, they are logical inconsistent and the detection is ended. This step aims at identifying information logical inconsistency in these observations when the stored knowledge is sufficient enough.

Step 4: Divide S^* into two parts S_1^* and S_2^* such that

$$S_1^* = \{\omega|_{J(C(\Omega(t)))} \mid \omega \in S^*\},$$
$$S_2^* = \{\omega|_{J(\omega(t)) \setminus J(C(\Omega(t)))} \mid \omega \in S^*\}$$

For S_1^*, let $S^* = S_1^*$, go to Step 3. For S_2^*, we cannot use the stored knowledge to detect logical inconsistency of observations in it. Hence, these observations in S_2^* are treated as new data and detected by the related data inconsistency methods as presented in Bruni (2004); Russell and Norvig (1995); Beliakov and Warren (2001). For each observation $s \in S_2^*$, if there exists an observation in S_2^* which is inconsistent, then the observations S^* are inconsistent. In this situation, the detection then goes to Step 5. However, when the stored knowledge is insufficient for detecting all observations, current process is applied. This step is based on the fact that if a part of these observations are logical inconsistent, then all of them as a whole must be logical inconsistent.

Step 5: For any consistent observation $s \in S_2^*$, construct $C(C(\Omega(t)), s)$ and add it to $\Omega(t)$. This step is an additional work, which updates the stored knowledge to preserve the completeness and effectiveness of a knowledge base. The updated knowledge base is still consistent.

Step 6: Explain conclusion and end.

By above steps, we can implement the information logical inconsistency detection for real-time observations.

4.3.4 Cases study of warning information inconsistency detection

In this section, the effectiveness and possible applications of the proposed SCLID method are illustrated through two examples.

First, we use the SCLID method for single object with two opposite states.

Example 4.3.1. *Power station's safety is a guarantee for the safety of a nuclear reactor. Monitoring a power station's running is an important task. Suppose a power station has a monitoring system which has 14 lookouts, p_j, $j = 1, 2, \ldots, 14$, distributed in different places. Each lookout will report the states of its local place every hour. Suppose a knowledge base $\Omega(t)$ of the power station's function is shown in Table 4.8, which includes 10 pieces of knowledge ω_i, $i = 1, 2, \ldots, 10$. Moreover, suppose that the reported states of these outlooks have two possible values, i.e., 1 (for abnormal* *function) and* 0 *(for normal function).*

Table 4.8: An example knowledge base

No.	Knowledge
ω_1	$\{(p_1 = 1, p_2 = 1, p_5 = 1, p_6 = 1)\}$
ω_2	$\{(p_2 = 1, p_{14} = 1)\}$
ω_3	$\{(p_6 = 1, p_{10} = 1)\}$
ω_4	$\{(p_3 = 1, p_4 = 1, p_7 = 1)\}$
ω_5	$\{(p_{10} = 1, p_{14} = 1)\}$
ω_6	$\{(p_7 = 1, p_{10} = 1, p_{11} = 1)\}$
ω_7	$\{(p_8 = 1, p_{11} = 1)\}$
ω_8	$\{(p_8 = 1, p_7 = 1)\}$
ω_9	$\{(p_{10} = 1, p_{12} = 1)\}$
ω_{10}	$\{(p_{10} = 1, p_{13} = 1)\}$

Let $S^* = \{(p_1 = 1,\ p_2 = 1,\ p_3 = 1,\ p_4 = 1)\}$ *be a set of real observations. We want to know if the power is normally running. To do that, we will use the SCLID method to check if these real observations are logical consistent or not.*

Using the presented SCLID method, we have:

Step 1: Obviously, $C(S^*)$ *isn't empty. Go to Step 2.*

Step 2: By coupling operation on the knowledge bases $C(\Omega(t))$*, we obtain a piece of knowledge:*

$$\{(p_1 = 1, p_2 = 1, p_3 = 1, p_4 = 1, p_5 = 1, p_6 = 1, p_7 = 1,$$
$$p_8 = 1, p_{10} = 1, p_{11} = 1, p_{12} = 1, p_{13} = 1, p_{14} = 1)\}.$$

As $J(S^*) \subseteq J(C(\Omega(t)))$*, then go to Step 3.*

Step 3: By Definition 4.3.4, we have

$$(C(\Omega(t)))|_{\{1,2,3,4\}} = \{(p_1 = 1, p_2 = 1, p_3 = 1, p_4 = 1)\}. \tag{4.22}$$

So, $S^* = C(\Omega(t))|_{\{1,2,3,4\}}$*. The observations are consistent, which means the power station is functioning abnormally. Stop.*

Now, let $S^* = \{(p_1 = 1,\ p_2 = 1,\ p_3 = 1,\ p_9 = 1)\}$*. By the proposed SCLID method, we have*

Step 2: $J(S^*) \nsubseteq J(C(\Omega(t)))$*. Then go to Step 4.*

Step 4: Dividing S^* *into* $S_1^* = \{(p_1 = 1, p_2 = 1, p_3 = 1)\}$ *and* $S_2^* = \{(p_9 = 1)\}$*. For* S_1^**, goto Step 3. For* S_2^**, without loss of generality, suppose it is consistent by the rule-map technique, then go to Step 5.*

Step 3: For S_1^**, we know it is consistent. Hence, the power station is function abnormally.*

Step 5: Because the S^* *is a set of consistent observations, we shall update our knowledge by coupling* $\{(p_9 = 1)\}$ *and* $C(C(\Omega(t)))$ *and have*

$$\{(p_1 = 1, p_2 = 1, p_3 = 1, p_4 = 1, p_5 = 1, p_6 = 1, p_7 = 1,$$
$$p_8 = 1, p_9 = 1, p_{10} = 1, p_{11} = 1, p_{12} = 1, p_{13} = 1, p_{14} = 1)\}.$$

Step 6. End.

From this example, we can see that the presented SCLID method is very effective as it only needs to detect a small part of all possible state combinations of related objects. This feature is suitable for a real problem because it can reduce the searching space and save the searching time by avoiding detection for insignificant combinations.

In the next example, we use the SCLID method for objects with multiple states. In this situation, a piece of knowledge may cover multiple state combinations.

Example 4.3.2. *Suppose another warning system monitoring the changes of three objects,* A*,* B*, and* C*. Each object can take observation values from* $\{$*slow (1), medium (2), fast (3)*$\}$*. Let* $r(A, B)$ *be the knowledge "*A*'s change is greater than* B*'s change." Now we have a knowledge base* $\Omega = \{\omega = r(A, B),\ \phi = r(B, C)\}$ *and a set of* *real-time observations* $S^* = \{(A = 2, B = 2, C = 1)\}$. *Hence we have*

$$\omega = \{(A = 2, B = 1), (A = 3, B = 1), (A = 3, B = 2)\}$$

$$\phi = \{(B = 2, C = 1), (B = 3, C = 1), (B = 3, C = 2)\}.$$

Using the SCLID method, we have the following steps to detect logical inconsistency for the real-time observations.

Step 2: By coupling these two pieces of knowledge, we have

$$C(\Omega(t)) = \{(A = 3, B = 2, C = 1)\}. \tag{4.23}$$

Because $J(S^*) = J(C(\Omega(t)))$, *go to Step 3.*

Step 3: Notice that $(C(\Omega(t)))|_{\{A,B,C\}} = \{(A = 3, B = 2, C = 1)\}$ *and* $S^* \nsubseteq (C(\Omega(t)))|_{\{A,B,C\}}$, *therefore, the observations are logical inconsistent. Then go to Step 6 and stop.*

The presented SCLID method is not only be used for the real-time observations but also be used for detecting logical inconsistency in knowledge bases. By taking knowledge in a knowledge base as a set of observations, we can treat the knowledge base as being generated from an empty knowledge base. Hence, applying the proposed method, we can detect the logical inconsistency of the knowledge in the knowledge base.

Continuing Example 4.3.2, suppose we have the third piece of knowledge $\psi = r(C, A)$. We have

$$\psi = \{(C = 2, A = 1), (C = 3, A = 2), (C = 3, A = 1)\}. \tag{4.24}$$

Now $C(\Omega) = \mho$, which means the knowledge base is inconsistent.

By taking this advantage, we can use the SCLID method to detect logical inconsistency in both real-time and stored knowledge for a warning system. Obviously, this can improve the facility and function of a warning system.

Chapter 5 Decision Information Integration

Information integration (also known as information fusion) is to combine information obtained from different sources to get a comprehensive picture of the event or situation considered. It is a very important step for implementing a people-centred warning system and supporting decision-making in a complex and dynamic environment.

This chapter focuses on information integration problems with qualitative or subjective nature based on a real application in nuclear safeguards information management field.

5.1 Introduction of nuclear safeguards information management

Safeguards information management of nuclear materials, according to the definition from the International Atomic Energy Agency (IAEA), aims at the timely detection and deterrence of non-peaceful or unclear diversion of nuclear materials in a state or an area. The used information for evaluating a state's nuclear material and nuclear-related activities is mainly collected from three information sources, i.e., the state (e.g., state declaration), the IAEA (e.g., verification reports), and open sources (e.g., media and open databases) (Liu *et al.*, 2002). Among them, the amount of qualitative information becomes enormous. Hence, there is an urgent need to establish a quality theoretical framework and practical method for integrating heterogeneous information and supporting accurate awareness of nuclear-related activities.

To develop a method to integrate qualitative information for nuclear safeguards information management, three aspects of the real settings of nuclear safeguards are commonly considered, i.e., the hierarchy of known nuclear activities and indicators, the different strengths of indicators, and the representation of qualitative information.

5.2 The physical model and the extended physiscal model

5.2.1 The physical model for nuclear safeguards information

The hierarchy of known nuclear process activities and indicators is depicted in the IAEA Physical Model (PM) (Liu *et al.*, 2002; Liu and Morsy, 2007) for example in the nuclear fuel cycle. The PM includes the relevant nuclear-related activities from source material acquisition to the production of materials for nuclear weapons. The relationships between every known nuclear activities and the indicators are identified. These relationships are described at three levels from top to bottom, i.e., the phase level, the process level, and the technology level (Maschio, 2007). A basic PM structure is shown in Figure 5.1. As Figure 5.1 has shown that 1) each indicator at the technology level has a dependent indicator at the process level and each indicator at the process level has a dependent indicator at the phase level; and 2) all indicators are organised in a tree-like hierarchy. The PM provides convenience for organizing and analysing the safeguards relevant information and has been used for evaluating undeclared nuclear activities or misuse of declared facilities (Liu *et al.*, 2002).

5.2.2 Strengths of nuclear activity indicators

In PM, an indicator's strength describes the degree of dependency between it and its dependent indicator. More detailed, the strength of an indicator at the technology

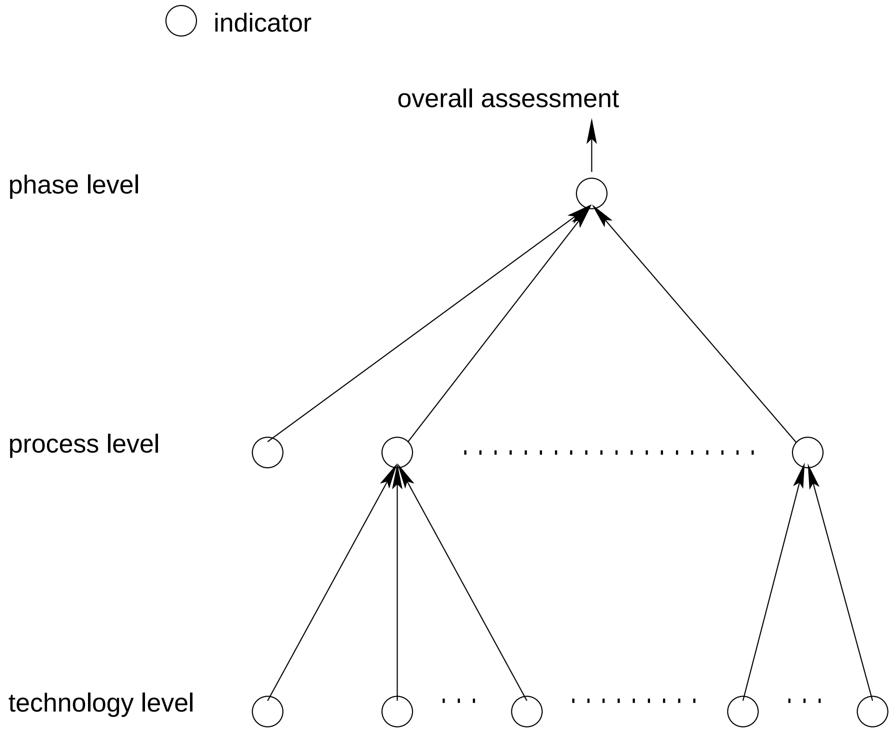


Figure 5.1: A general structure of the IAEA Physical Model

level means that if the technical activity, which is associated with the indicator, is used, to what extent a corresponding processing activity at the process level, which uses this technique, is being conducted.

Traditionally, three kinds of strengths are noted in nuclear safeguards information management, i.e., strong, medium, and weak (Liu *et al.*, 2002; Maschio, 2007). An indicator x is said to be a strong indicator of an indicator y if the processing activity associated with y implies and is implied by the processing activity associated with x. While, an indicator x is a medium indicator of y if the processing activity associated with y implies and may be implied by the processing activity associated with x; an indicator x is a weak indicator of y when the processing activity associated with y may imply and may be implied by the processing activity associated with x.

5.2.2.1 An analysis of strengths from logics

The dependent relation between known nuclear process activities can be analyzed from the viewpoint of logic as follows. Let $P(x)$ and $P(y)$ be the activities associated with indicators x and y, respectively. An indicator x is a strong indicator of a process activity $P(y)$ means the following logical formula F

$$P(x) \Rightarrow P(y) \quad \text{AND} \quad P(y) \Rightarrow P(x), \tag{5.1}$$

always takes the truth value 1 (or True), i.e., F is a tautology. In classical logic, F is a tautology if and only if both $P(x) \Rightarrow P(y)$ and $P(y) \Rightarrow P(x)$ take the truth value 1. Furthermore, both $P(x)$ and $P(y)$ take the truth value 1. More generally, $P(x) \Rightarrow P(y)$ takes the truth value 1 implies that

$$v(P(x)) \leqslant v(P(y)), \tag{5.2}$$

where $v(P(x))$, $v(P(y))$ are the truth values of $P(x)$ and $P(y)$ respectively. Similarly, the following equation holds

$$v(P(y)) \leqslant v(P(x)) \tag{5.3}$$

because $P(y) \Rightarrow P(x)$ always takes the truth value 1. Hence for a strong indicator, we have from Eq. (5.2) and Eq. (5.3)

$$v(P(y)) = v(P(x)). \tag{5.4}$$

For a medium indicator, a corresponding logical formula, which always takes truth value 1, can be expressed as

$$P(y) \Rightarrow P(x), \tag{5.5}$$

Therefore

$$v(P(y)) \leqslant v(P(x)). \tag{5.6}$$

As for a weak indicator, we cannot establish a definitive logical formula to describe the relation between the weak indicator and a process activity due to their loose link.

5.2.2.2 An analysis of strengths from possibility

Moreover, we can analyse the strengths of indicators from the possibility viewpoint. Without loss of generality, let the truth value $v(P(x))$ be the possibility of the fact that the process activity associated with indicator x is being conducted. We draw Eq. (5.4) and Eq. (5.6) in Figure 5.2 and use the figure to analyse the differences in strengths and their effects on a process activity.

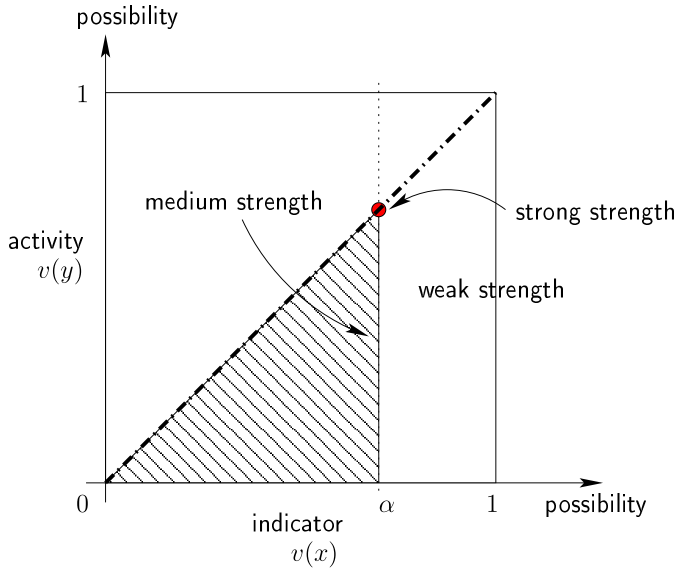


Figure 5.2: Possibilities of activities associated with indicators

Suppose a process activity $P(x)$ is being conducted with possibility α and y is the dependent indicator of x. If x is a strong indicator of y, then by Eq. (5.4), it is rational to assume that the process activity $P(y)$ is also being conducted with the possibility α. If x is a medium indicator of y, it is still rational to assume that the process activity $P(y)$ is being conducted with the possibility in a definitive interval $[0, \alpha]$ by Eq. (5.6), although we cannot definitively determine the value. However, if x is a weak indicator of y, it will be more difficult to find an approximate interval for the possibility of $P(y)$. These facts indicate that indicator with stronger strength will provide more confirm information. Therefore, a rational hypothesis is given below which will be used as the basis of the proposed qualitative information integration (QII) method in this section.

Hypothesis Let x_1, x_2, and x_3 be three indicators with same dependent indicator y and the strengths of them be “strong,” “medium,” and “weak”, respectively. If the possibilities of three nuclear activities $P(x_1)$, $P(x_2)$, and $P(x_3)$, which are being conducted, are the same, then the possibilities p_1, p_2 and p_3 of the hypothesis that $P(y)$ is being conducted, which are derived respectively from x_1, x_2 and x_3, satisfy

$$p_1 \geqslant p_2 \geqslant p_3. \tag{5.7}$$

5.2.3 The extended physical model

The basic PM is simple in structure; but lacks some important features in real applications. Observing Figure 5.1, It is noted that hierarchy of indicators in the basic PM is a tree, i.e., each indicator at a lower level has a unique dependent indicator at a higher level. Obvious merits of this hierarchy include its conciseness in structure and its popularity in applications. However, such a hierarchy does not satisfy the real setting of applications including nuclear safeguards assessments. For example, a technology may be used in different processes in nuclear safeguards assessments.

Based on the basic PM model, an extended PM model, called XPM, is developed and given in Figure 5.3. The XPM has two main extensions of the PM. Firstly, the XPM adds an information source level below the technology level. The information source level provides supporting information to the technology level. Secondly, each indicator at the technology level can provide supporting information to multiple indicators at the process level. The XPM is an extension of the PM model; but is not limited to this.

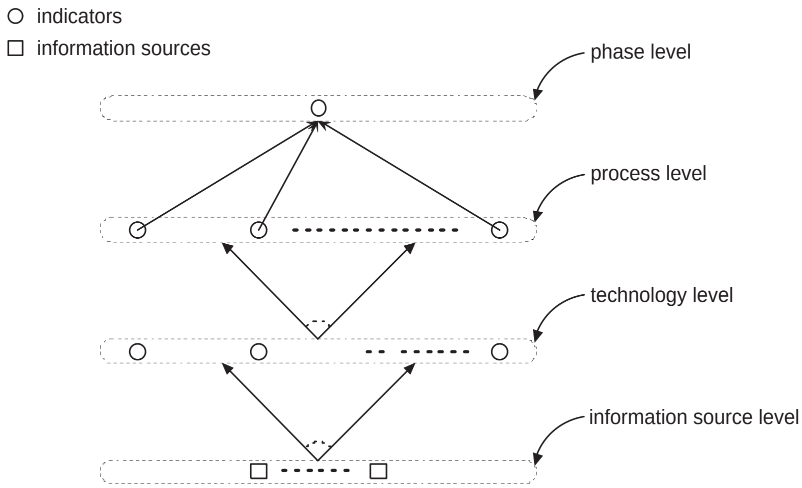


Figure 5.3: An extended PM model (XPM)

5.3 Qualitative information representation and process

A large part of the collected information for nuclear safeguards is with qualitative features. Because information is collected from different information sources with various natures, it is hard to give a common representation for all kinds of qualitative information. Qualitative information is commonly expressed as linguistic terms. Noting that the meanings or semantics of a piece of qualitative information often cover multiple linguistic terms, this study, therefore, represents a piece of qualitative information as a possibility distribution (PD) over the selected linguistic terms.

To represent qualitative information as possibility distributions, we need to resolve two basic problems. The first problem is how to transfer indicator-oriented information to activity-oriented information. The second problem is how to transfer information to possibility distribution. Because these two problems link closely to each other, we use the following strategy to solve them simultaneously. We firstly interpret the linguistic terms as descriptions of a process activitys conducting states. For instance, a process activity may be described as “not being conducted, “highly being conducted, “being conducted or some other descriptions. Then for a piece of information about an indicator, we transfer the information of the indicator to a possible description of its associated process activitys states. Suppose, for instance, an indicator records the number of a specific device which is used in the process activity associated with the indicator and the number of that device should be limited to n_p in normal situation. If the number of that device is about n_r now, an estimation about the conducting state of the associated activity can be obtained by comparing n_r and n_p. Hence, a possibility distribution over the descriptions of the associated activitys conducting states can be obtained. By this way, the information n_r about indicator is transferred to a possibility distribution over an activitys conducting states.

5.4 A qualitative information integration method

In this section, an XPM-based qualitative information integration (QII) method is presented for qualitative information integration in nuclear safeguards information management problem discussed in Section 5.1.

5.4.1 An overview

The proposed QII method consists of two successive stages (Figure 5.4) as shown in Table 5.1. Detailed processing in each stage (and steps) is described in the following sections. In order to clearly refer indicators at different levels, the indicators at the process level are called factors and the indicator at the phase level is called a judgment. Suppose a judgment A has m possible states con_1, con_2, ..., con_m. Each state is represented by a linguistic term in S. The output of the presented method is a synthesized possibility distribution (SPD) on all possible states of A.

Stage 1 implements information transformation. Each piece of indicator-oriented information collected from multiple sources is converted to a possibility distribution at the technology level. This transformation is conducted with the help of domain experts. Because each indicator depends on multiple information sources and those

Table 5.1: Outlines of the QII method

QII method
stage 1: information transformation
input: collected information
functions: get information from sources and transform to PDs on connecting indicators
output: SPDs on indicators (technology level)
stage 2: qualitative information integration
input: SPDs on indicators
function: integrate SPDs on indicators (technology level) to get SPD on process activity (phase level)
step 1 (indicator-factor grouping aggregation) : integrate SPDs on indicators (technology level) to get SPD on factors (process level)
input: SPDs on indicators
output: SPDs on factors
details:
1) divide indicators into several groups by their strengths
2) (indicator-group aggregation) aggregate SPDs on indicators belong to same group
3) (group-factor aggregation) aggregate SPDs on groups
step 2 (factor-judgment grouping aggregation): integrate SPDs on factors (process level) to get SPD on process activity (phase level)
input: SPDs on factors
output: SPDs on process activity
details:
1) divide factors into several groups by their strengths
2) (factor-group aggregation) aggregate SPDs on factors belong to same group
3) (group-judgment aggregation) aggregate SPDs on groups
output: SPDs on process activity

information sources are with different reliabilities, the transferred information for the same indicator will be integrated to generate an SPD as inputs for process in Stage 2.

Stage 2 implements qualitative information integration about indicators at the technology level and the process level and generates an SPD for the judgment A. The integration procedure is conducted with the help of aggregation operators and implication operators. The input of Stage 2 is a set of PDs for indicators at the technology level, which is generated in Stage 1.

For convenience, we will use the symbols in Table 5.2 in the following without other specifications.

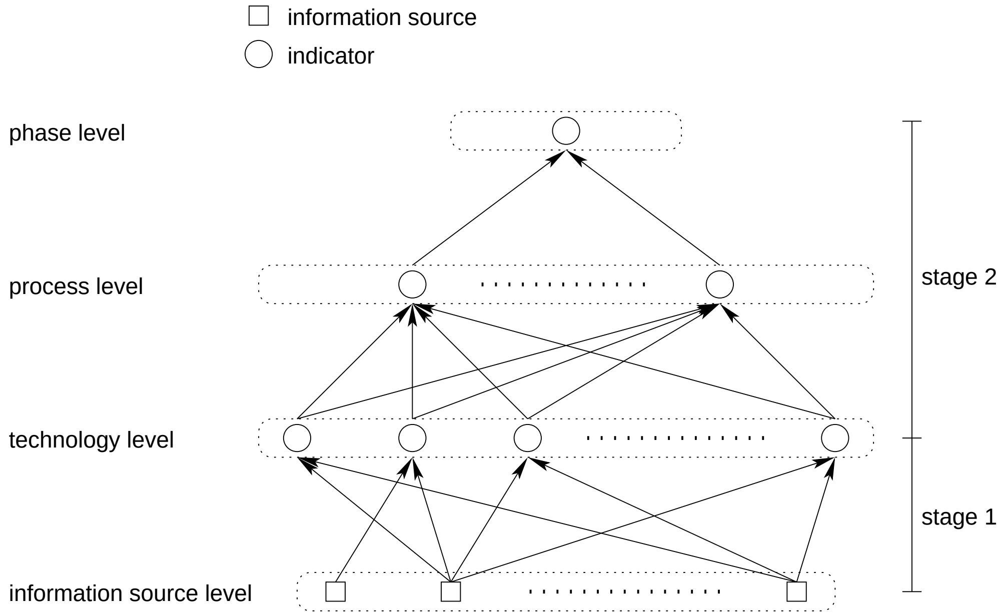


Figure 5.4: Information integration based on XPM

Table 5.2: Symbols and meanings used in the QII method

Symbol	Meaning
$\text{CON} = \{\text{con}_1, \cdots, \text{con}_m\}$	possible conclusions of a judgment
$\text{IND} = \{\text{ind}_1, \cdots, \text{ind}_n\}$	indicators (criteria)
$\text{INF} = \{\text{inf}_1, \cdots, \text{inf}_k\}$	collected information
$\Omega = \{\omega_1, \cdots, \omega_p\}$	strengths of indicators
$\Gamma = \{\gamma_1, \cdots, \gamma_q\}$	reliability of information

5.4.2 Information transformation

In Stage 1, the QII method will complete two tasks: 1) transfers each piece of original information to a set of possibility distributions over possible conducting states of process activities which are associated with its dependent indicators; 2) generates an SPD for each indicator at the technology level. The first task is conducted with the help of domain experts and the second task is implemented through selecting approximate aggregation operations.

Suppose inf_i is a piece of information which supports indicator ind_j. Let f_{ij} be a transformation function determined by indicator ind_j. Then a PD on CON can be obtained (denoted by p_{ij}):

$$p_{ij} = (a_{ij1}, \cdots, a_{ijm}). \tag{5.8}$$

where a_{ijk} is the possibility of conclusion con_k occurring based on indicator ind_j given information inf_i, $k = 1, \ldots, m$. Therefore, a possibility distribution matrix $M_j = (p_{j_1 j}, \cdots, p_{j_s j})^T$ is obtained for each indicator ind_j. An SPD for indicator ind_j is obtained by

$$p_j = (\gamma_{j_1}, \cdots, \gamma_{j_s}) \circ M_j, \tag{5.9}$$

where $\circ$ is an aggregation operator selected for indicator ind_j, and γ_{j_i} is the reliability of information inf_i. The obtained SPD is the overall awareness of conducting state of the process activity which is associated with indicator ind_j.

Naturally, an important problem is that how to select an aggregation operator in Eq. (5.9). We will discuss this problem in Section 5.4.4.

Similarly, we can calculate the SPDs for the other indicators at the technology level. Once these SPDs are obtained, the process turns to Stage 2.

5.4.3 Information integration

After the process at Stage 1, each indicator is associated with an SPD which can be treated as awareness of conducting states of the associated process activity. Based on these SPDs (noted that they are all possibility distributions), the QII method will complete two tasks at Stage 2: 1) For each factor at the process level, PDs for its supporting indicators at the technology level are integrated through an indicator-factor grouping aggregation strategy in order to obtain an SPD for it; and 2) For the final judgment, PDs for its supporting factors at the process level are integrated through a factor-judgment grouping aggregation strategy in order to get an SPD on its all possible states.

Here, we mainly focus on the process in Step 1 because the process in Step 2 is similar to it.

The indicator-factor grouping aggregation includes the indicator-group aggregation and the group-factor aggregation which are illustrated below.

The indicator-group aggregation is conducted in the following way. Indicators at the technology level which support the same factor at the process level will be divided into several groups in terms of their strengths. In each group, an aggregation operator is used on all PDs for indicators in the group to generate an SPD for the group.

The group-factor aggregation is conducted in the following way. A second aggregation operator is used on SPDs for all indicator groups to generate an SPD for the factor.

Suppose ind_{g_1}, ind_{g_2}, $\cdots$, ind_{g_t} are indicators belonging to the same group g, in which all indicators have the same strength. At the indicator-group aggregation, the aggregation result on g is calculated by

$$p_g = \mathrm{Agg}_g(p_{g_1}, p_{g_2}, \ldots, p_{g_t}), \tag{5.10}$$

where Agg_g is a selected aggregation operator for group g, p_g is the SPD for group g, and p_{g_i}, $i = 1, \ldots, t$, is the PD for indicator ind_{g_i}. The process, then, enters the group-factor aggregation. Suppose the supporting indicators of a factor f are grouped into r groups $g_1^{(f)}$, $g_2^{(f)}$, ..., $g_r^{(f)}$, then the SPD for factor f is conducted by

$$p_f = \mathrm{Agg}_f(h(p_{g_1^{(f)}}, \omega_{g_1^{(f)}}), \ldots, h(p_{g_r^{(f)}}, \omega_{g_r^{(f)}})) \tag{5.11}$$

where f represents a factor, h is a function which is used to combine information in a PD and its strength, and Agg_f is a selected aggregation operator. (Similarly, the problem of aggregation operator selection arises and we will discuss the solution in Section 5.4.4.)

After getting SPDs for all factors at the process level, the QII method repeats the similar factor-judgement grouping aggregation strategy for all factors to get an SPD for the final judgment.

5.4.4 Aggregation operators selection

An important issue in the proposed QII method is how to select a suitable aggregation operator in grouping aggregation.

As the analysis above, the strength of an indicator reflects an implied logical relationship between two related process activities. For example, a strong indicator is a necessary and sufficient condition of a process activity; while a medium indicator is a necessary condition of a process activity. From the viewpoint of logic, the relationship between two process activities $P(x)$ and $P(y)$ can be extended as

$$(v(P(x)) \to v(P(y)) = \alpha) \text{ AND } (v(P(y)) \to v(P(x)) = \beta), \tag{5.12}$$

where $v(P(x))$, $v(P(y))$ are estimated possibilities of the two activities, respectively; and α, β are two parameters indicating the correctness of these two claims.

Let us consider Eq. (5.12) in the conventional two-valued logic. Suppose $\alpha = 1$, $\beta = 1$, and interpret AND as the conjunction operator, then in this setting it is easy to see that both $v(P(x)) \to v(P)$ and $v(P(y)) \to v(P(x))$ must take truth value 1 (means **True**). Therefore we have $v(P(x)) = v(P(y))$ which means both process activities occur simultaneously. Hence, x is a strong indicator of y. If let $\alpha = 0$ and $\beta = 0$, then in this setting, we cannot find a specific relationship between $v(P(x))$ and $v(P(y))$, so the indicator x can be seen as a weak indicator of y. Let $\alpha = 0$ and $\beta = 1$. In this case, if $v(P(y)) = 1$, then $v(P(x)) = 1$. But the converse may not hold. So, x is a medium indicator of process activity y. An interesting conclusion drawn from the above analysis is that the selection of α and β can approximately illustrate different kinds of strengths.

Now, we expand this discussion and suppose α, $\beta \in [0, 1]$. Let x, y, z be indicators associated to process activity P with strengths ω_x, ω_y, and ω_z respectively, and suppose $\omega_x \geqslant \omega_y \geqslant \omega_z$. Given a possibility of P, we have intuitively that

$$v(P(x)) \geqslant v(P(y)) \geqslant v(P(z)), \tag{5.13}$$

where $v(P(x))$, $v(P(y))$, and $v(P(z))$ are the possibilities of $P(x)$, $P(y)$ and $P(z)$, respectively. Hence, we have from the viewpoint of logic that

$$v(P(y)) \rightarrow v(P(x)) \geqslant v(P(y)) \rightarrow v(P(y)) \geqslant v(P(y)) \rightarrow v(P(z)). \tag{5.14}$$

where $\rightarrow$ is an implication operator. On the other hand, suppose $v(P(x)) = v(P(y)) = v(P(z))$. Intuitively, we have

$$v_x(P) \geqslant v_y(P) \geqslant v_z(P), \tag{5.15}$$

because x is more important than y and z and y is more important than z. By Eq. (5.15), we have

$$v(P(x)) \rightarrow v_x(P) \geqslant v(P(y)) \rightarrow v_y(P) \geqslant v(z) \rightarrow v_z(P). \tag{5.16}$$

Hence $\omega_x \geqslant \omega_y \geqslant \omega_z$ can be described by equations (5.14) and (5.16).

Recalling equations (5.12), (5.14) and (5.16), we noted that the strength of an indicator increases with in the increasing of the value of α and β. Hence, we redefine a new strength by α and β as follows:

$$\omega = (\alpha, \beta). \tag{5.17}$$

Moreover, we define the order on strengths as

$$\omega_x \geqslant \omega_y \text{ if and only if } \alpha_x \geqslant \alpha_y \text{ and } \beta_x \geqslant \beta_y. \tag{5.18}$$

Thus, we are able to describe the relation among strengths by Figure 5.5, in which we show the positions of the three typical strengths. It is easy to know that the strength becomes stronger when α and β increase along the arrow's direction.

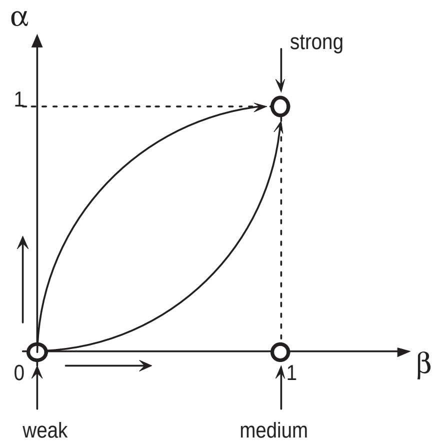


Figure 5.5: Strengths of indicators

However, the relationship in Eq. (5.16) is more essential than that in Eq. (5.14) in the evaluation of information because $v(P(y))$ is not known in advance. Hence the selection of aggregation operators is mainly based on $v(\cdot) \to v(P(y))$. In the following, we refer a strength of the indicator to α only.

According to the change trend of a strength degree, a principle of selecting an aggregation operator in information integration can be obtained. Let ω_x and ω_y be two strengths and suppose $\omega_x \geqslant \omega_y$. In order to get the SPD in terms of indicators x and y, we select two aggregation operators Agg_x and Agg_y such that

$$\mathrm{Agg}_x(p) \geqslant \mathrm{Agg}_y(p) \text{ for the same PD } p.$$

For example, if the strength of indicator x is stronger than that of y, we can select the max-type aggregation operator for x and the min-type aggregation for y. This selection strategy is obtained based on the consideration that a stronger indicator should provide more alertness than the weaker one. We apply this strategy to the aggregation in Eq. (5.9) and Eq. (5.10).

For the aggregation in Eq. (5.11), another strategy is applied on the consideration that a potential logical inference exists between indicators. It is known that a stronger indicator is required to produce a more accurate estimation about a process activity; at least a higher alertness is required even if an accuracy requirement cannot be fully satisfied. Hence, under the same condition, the possibility of a process activity obtained from a stronger indicator should be larger than that from a weaker indicator. Considering the logical relationship between an indicator and a process activity, we use the idea of the transformation function in Ruan *et al.* (2003); Yager (1981, 1993, 1994) to implement the aggregation in Eq. (5.11) as follows:

Step1: Let $\rightarrow$ be an implication operator, ω_g be the strength of indicator group g, and p_g the SPD of this group obtained by Eq. (5.10). For each $p_{gi} \in p_g$, a PD $\tilde{p}_{gi}$ is obtained by

$$\tilde{p}_{gi} = (\alpha_g \rightarrow p'_{gi})', \tag{5.19}$$

where $'$ is a complementary operation induced by an implication operator $\rightarrow$ ($x' = x \rightarrow 0$). Then we get a set of PDs $\tilde{p}_{g_{\sigma_1}}, \tilde{p}_{g_{\sigma_2}}, \cdots, \tilde{p}_{g_{\sigma_r}}$.

Step 2: To aggregate $\tilde{p}_{g_{\sigma_1}}$, $\tilde{p}_{g_{\sigma_2}}$, $\cdots$, $\tilde{p}_{g_{\sigma_r}}$ for a factor, we implement an OWA aggregation by

$$p_{f1} = \sum_{j=1}^{r} \omega_{g_{\sigma_j}} b_j, \tag{5.20}$$

where $\omega_{g_{\sigma_j}}$ is the weight of PD $\tilde{p}_{g_{\sigma_j}}$, which is determined by the strength of the indicator ind_{g_i}, and b_j is the j-th largest of $\tilde{p}_{g_{\sigma_i}}$.

Therefore, we complete the aggregation in Eq. (5.11).

Remark 5.4.1. *In Step 2 above, we use the OWA aggregation operator for example. In real applications, the aggregation operator can be in different forms. The selection* *of an appropriate aggregation operator in real application is still an open problem (Beliakov and Warren, 2001; Beliakov and Calvo, 2008).*

5.5 A case study on nuclear safeguards information for warning generation

5.5.1 Brief illustration and settings

In this section, we apply the proposed QII method to an example recited from Liu *et al.* (2002). The example is briefly described as below.

That conducting a specific process for gaseous diffusion enrichment is an important activity for production of highly enriched uranium (HEU). Associated with the process activity, a set of specific indicators are identified, which are shown in Table 5.3, and three strengths are assigned to them, i.e., strong, medium, and weak. The numeric values for three kinds of strengths are set as 9 (strong), 3 (medium), and 1 (weak). The assessments from four experts (information sources) are listed in Table 5.4 (Liu *et al.*, 2002) in which integers represent indexes of terms taken from a linguistic term set $S = \{s_0 =$ impossible, $s_1 =$ almost impossible, $s_2 =$ slightly possible, $s_3 =$ quite possible, $s_4 =$ possible, $s_5 =$ high possible, $s_6 =$ absolutely possible$\}$. The importance (reliability) of each expert (information source) is also expressed by terms taken from S and is set as $E_1 = r_3$, $E_2 = r_5$, $E_3 = r_4$, and $E_4 = r_2$, where r_i in $W = \{r_0 =$ none, $r_1 =$ very low, $r_2 =$ low, $r_3 =$ medium, $r_4 =$ high, $r_5 =$ very high, $r_6 =$ perfect$\}$. The final judgment (P) is about "whether it is conducting a specific process Gaseous diffusion enrichment".

Table 5.3: Specific indicators of gaseous diffusion enrichment

ID	Denomination	Type	Strength
258	expansion bellows	dual-use equipment	weak
259	gasket, large	dual-use equipment	weak
261	gas blowers for UF6	especially designed equipment	medium
262	rotary shaft seal	especially designed equipment	medium
265	compressor for pure UF6	especially designed equipment	strong
266	gaseous diffusion barriers	especially designed equipment	strong
267	heat exchanger for cooling pure UF6	especially designed equipment	strong
268	feed system/product and tails withdrawal	especially designed equipment	weak
269	header piping system	especially designed equipment	weak
271	chlorine trifluoride	non-nuclear material	medium
272	nickel power, high purity	non-nuclear material	medium
273	aluminum oxide powder	non-nuclear material	weak
276	large electrical switching yard	non-nuclear material	weak
277	large heat increases in air or water	other	weak
279	large specific power consumption	other	weak

source: Liu *et al.* (2002)

Table 5.4: Evaluation for process — gaseous diffusion enrichment

E_1	E_2	E_3	E_4	Denomination	Strength
6	6	6	5	expansion bellows	weak
2	3	5	3	gasket, large	weak
3	2	3	6	gas blowers for UF6	medium
4	3	5	3	rotary shaft seal	medium
4	2	4	6	compressor for pure UF6	strong
6	5	4	6	gaseous diffusion barriers	strong
5	3	6	6	heat exchanger for cooling pure UF6	strong
1	3	2	4	feed system/product and tails withdrawal	weak
5	3	6	4	header piping system	weak
3	2	5	4	chlorine trifluoride	medium
2	2	3	4	nickel power, high purity	medium
2	2	2	3	aluminum oxide powder	weak
3	6	5	5	large electrical switching yard	weak
6	3	6	4	large heat increases in air or water	weak
4	3	5	6	large specific power consumption	weak

source: Liu *et al.* (2002)

5.5.2 Decision information transformation

Because the current settings of the example are not coinciding with the proposed QII method, some transformation is made to these settings.

First, let the final assessment be a set of seven conclusions (CON):

$$\begin{aligned}
\mathrm{con}_0 &:= \text{ impossible of } P \\
\mathrm{con}_1 &:= \text{ almost impossible of } P \\
\mathrm{con}_2 &:= \text{ slightly possible of } P \\
\mathrm{con}_3 &:= \text{ quite possible of } P \\
\mathrm{con}_4 &:= \text{ possible of } P \\
\mathrm{con}_5 &:= \text{ highly possible of } P \\
\mathrm{con}_6 &:= \text{ absolutely possible of } P,
\end{aligned}$$

and con_i corresponds to s_i, $i = 0, 1, \ldots, 6$.

Next we normalize the strengths of indicators such that

$$\omega_{strong} = 9/13, \quad \omega_{medium} = 3/13, \quad \omega_{weak} = 1/13.$$

Then we transfer the assessments from experts to PDs on "CON". Suppose a is an assessment from experts, which connects to an indicator "ind". Then the PD derived from a is

$$p_i = \max\{1 - \omega \cdot |i - a|, 0\}, \quad i \in \{0, 1, \ldots, 6\}. \tag{5.21}$$

where ω is the strength of indicator "ind". For example, 4 is the assessment of E_1 on indicator "compressor for pure UF6," then the possibility distribution derived from 4

is

$$p(\text{con}_0) = 0.0, \quad p(\text{con}_1) = 0.0, \quad p(\text{con}_2) = 0.0,$$
$$p(\text{con}_3) = 0.301, \quad p(\text{con}_4) = 1.0, \quad p(\text{con}_5) = 0.301, \tag{5.22}$$
$$p(\text{con}_6) = 0.0.$$

As for the reliabilities of experts, we translate them from a symbolic expression to a numeric value by setting

$$\gamma_i = \text{index}(r_i)/6, \quad i = 1, 2, 3, 4$$

where $\text{index}(r_i)$ is the index of linguistic term r_i, i.e., i. Therefore, $E_1 = 0.5$, $E_2 = 0.833$, $E_3 = 0.667$, $E_4 = 0.333$. Let $\Gamma = \{\gamma_i | i = 0, 1, \ldots, 6\}$.

Finally, notice that indicators are grouped into 4 types (factors), i.e., "dual-use equipment" (f_1), "especially designed equipment" (f_2), "non-nuclear material" (f_3), and "other" (f_4), we randomly assign the strengths to each factor as

$$w_1 = 0.3, \qquad w_2 = 0.4, \qquad w_3 = 0.2, \qquad w_4 = 0.1$$

due to the lack of background knowledge.

5.5.3 Solution

Based on the modified setting, we apply the presented method to this problem as follows.

At Stage 1, the information transformation is implemented through modifying the case settings as shown in Eq. (5.21). Then we need to integrate those PDs to generate SPDs for all indicators. For example, the matrix of possibility distribution associated with indicator "compressor for pure UF6" (ID=265) is

$$P_{265} = \begin{pmatrix} 0.0 & 0.0 & 0.0 & 0.301 & 1.0 & 0.301 & 0.0 \\ 0.0 & 0.301 & 1.0 & 0.301 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.301 & 1.0 & 0.301 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.301 & 1.0 \end{pmatrix}.$$

To obtain an SPD for this indicator, we apply P_{265} and Γ to Eq. (5.9) and select the max-type aggregation because the strength of this indicator is strong. Then we have

$$p_{265} = (0.0, 0.301, 1.0, 0.301, 1.0, 0.301, 1.0).$$

Similarly, we can obtain PDs on other indicators.

Table 5.5 shows the synthesized possibility distributions on all indicators.

Table 5.5: Synthesized possibility distributions (SPDs) for indicators

Denomination	Distribution
expansion bellows	(0.538, 0.615, 0.692, 0.769, 0.846, 0.923, 0.923)
gasket, large	(0.615, 0.692, 0.769, 0.846, 0.846, 0.769, 0.692)
gas blowers for UF6	(0.448, 0.641, 0.833, 0.667, 0.513, 0.359, 0.333)
rotary shaft seal	(0.257, 0.448, 0.641, 0.833, 0.641, 0.667, 0.513)
compressor for pure UF6	(0.000, 0.301, 1.000, 0.301, 1.000, 0.301, 1.000)
gaseous diffusion barriers	(0.000, 0.000, 0.000, 0.301, 1.000, 1.000, 1.000)
heat exchanger for cooling pure UF6	(0.000, 0.000, 0.301, 1.000, 0.301, 1.000, 1.000)
feed system/product and tails withdrawal	(0.692, 0.769, 0.846, 0.846, 0.769, 0.692, 0.615)
header piping system	(0.615, 0.615, 0.692, 0.769, 0.846, 0.846, 0.769)
chlorine trifluoride	(0.448, 0.641, 0.833, 0.641, 0.448, 0.269, 0.179)
nickel power, high purity	(0.448, 0.641, 0.833, 0.667, 0.513, 0.359, 0.205)
aluminum oxide powder	(0.769, 0.846, 0.923, 0.923, 0.846, 0.769, 0.692)
large electrical switching yard	(0.538, 0.615, 0.692, 0.769, 0.846, 0.846, 0.769)
large heat increases in air or water	(0.538, 0.615, 0.692, 0.769, 0.846, 0.846, 0.769)
large specific power consumption	(0.538, 0.615, 0.692, 0.769, 0.846, 0.846, 0.769)

When PDs on all indicators are obtained, we turn to Stage 2 to synthesize these PDs (information of indicators) for factors.

To implement this task, we first aggregate information of indicators supporting the same factor. Take the factor "especially designed equipment" (f_2) for example; this factor consists of seven indicators which can be divided into three indicator groups, i.e., three strong, two medium and two weak indicators. We take the max-type aggregation to synthesize PDs on strong indicators, min-type aggregation for weak indicators, and arithmetic mean for medium indicators. The aggregation results are listed in Table 5.6. Then we select the Kleene-Dienes-Łukasiewicz operator (Ruan *et al.*, 2003), i.e., $w \to a = 1 - w + w \cdot a$, and OWA aggregation to get SPD for factor (f_2) as below

$$p_{f_2} = (0.516, 0.516, 0.110, 0.110, 0.110, 0.110, 0.110). \tag{5.23}$$

Table 5.6: SPDs for the factor "especially designed equipment"

Strength of indicators	Distribution
strong	(0.000, 0.301, 1.000, 1.000, 1.000, 1.000, 1.000)
medium	(0.353, 0.545, 0.737, 0.750, 0.577, 0.513, 0.423)
weak	(0.615, 0.615, 0.692, 0.769, 0.769, 0.692, 0.615)

The SPDs for other factors can be obtained similarly and all these distributions are shown in Table 5.7.

Table 5.7: Synthesized possibility distributions for all factors

Factor ID	Distribution
f_1	(0.030, 0.024, 0.018, 0.012, 0.012, 0.006, 0.006)
f_2	(0.516, 0.516, 0.110, 0.110, 0.110, 0.110, 0.110)
f_3	(0.093, 0.060, 0.028, 0.057, 0.086, 0.112, 0.133)
f_4	(0.036, 0.030, 0.024, 0.018, 0.012, 0.012, 0.018)

According to the SPDs for all factors, the final conclusion for the problem is obtained by grouping aggregations. Because all factors have different strengths, each factor forms an individual group. Therefore, we only need to aggregate all SPDs for factors. For simplicity, we use the weighted sum and OWA aggregations to carry out the computation and make a comparison. By the weighted sum aggregation, the final possibility distribution is

$$p = (0.211, 0.213, 0.279, 0.279, 0.278, 0.277, 0.276),$$

which implies that the third and fourth conclusions are the evaluation result, which means “the possibility of process being conducted is approximate ‘slightly possible’.” While by using the OWA aggregation, we get

$$p = (0.220, 0.223, 0.279, 0.279, 0.278, 0.277, 0.276),$$

This is a similar result, i.e., the possibility of process being conducted is lower.

Chapter 6 Decision Information Prediction

This chapter discusses the decision information prediction problem, i.e., the multiple periodic factor prediction (MPFP) problem. This kind of problem exists widely in multi-sensor data fusion applications. Developing an effective prediction method should consider information of multiple periodically changing factors.

This chapter is organized as follows. Section 6.1 clarifies the MPFP problem by two examples and introduces the processing strategy in the study. The PSAO is introduced and its properties are discussed in Section 6.2. Section 6.3 proposes the PSAOP method for MPFP problems based on the PSAO. Two case studies, annual sunspot number prediction and bushfire danger rating prediction, are conducted in Section 6.4 to illustrate the detailed steps of the presented PSAOP method.

6.1 Introduction

Multi-sensor data fusion aims to combine data from multiple sensors, and related information from associated databases to achieve improved accuracies and inferences, which has widely applied in military and civilian applications (Hall and Llinas, 1997; Mitchell, 2007). Prediction problems which involve many periodically-changing factors commonly occur in those applications. The following applications are two typical examples.

Example 1: bushfire danger rating prediction. Australia is a bushfire-prone country. Predicting a bushfire danger rating in advance can reduce costs of damage and save people's lives. Australian fire authorities currently use a six-level fire danger rating system which is based on the Fire Danger Index (FDI). The FDI is calculated by using the data (observations and/or predictions) of four primary meteorological indicators, i.e., "maximum temperature", "efficient precipitation", "wind speed", and "relative humidity" (Lucas, 2010). It is well-known that the four indicators change seasonally. Hence, the same data for an indicator means different things at different times; for instance, a 20-degree temperature means a cool day in summer and a warm day in winter. This phenomenon indicates that data is of semantic uncertainty and periodicity. When using this kind of date to predict a possible FDI, the prediction method should handle them simultaneously.

Example 2: Customer churn management. Retaining churning customers is a challenging issue for telecommunication companies wanting the advantage over other telecommunication service providers in a competitive market. Potential churning customers have some similar behaviour patterns which are predictable through some deliberately selected drivers (variables) (Huang *et al.*, 2010). Raw data of these drivers is commonly numeric values, which is hard to use directly for decision-making. Linguistic descriptions, as a summarizing form of the raw data, is very understandable and can be used for decisions. However, it will introduce semantic uncertainty. Moreover, raw data is often used to extract the common behaviour patterns of a segment of likely churning customers. These patterns sometimes have unidentifiable periodicity, such as the patterns for "aggregated call-details" or the "bill and payment history". In order to retain likely churning customers, a decision made based on such data should also handle the semantic uncertainty and periodicity, simultaneously.

This kind of prediction problem is very common in practices. Although it can be categorised in the multi-sensor data fusion research fields (Hall and Llinas, 1997), this study refers it to the multiple periodic factor prediction (MPFP) problems to emphaise its specific characteristic, as listed below, and avoid misleading information. Bearing the above examples in mind, we summarise the common characteristics of MPFP problems as follows. Firstly, the prediction problem involves multiple periodically changing factors. These factors each may have their respective prediction models, eg., there are forecasting models for temperature (Chen and Hwang, 2000; Lee *et al.*, 2006). Secondly, the data of each factor is expressed in a time series or a fuzzy time series (Song and Chisson, 1993). Thirdly, a prediction is made based on aggregation of the data (observations/predictions) of these factors. The aggregation procedure needs to simultaneously process the semantic uncertainty and periodicity in the data.

Due to these characteristics, a remarkable difficulty in solving an MPFP problem is how to handle simultaneously the semantic uncertainty and periodicity in the data of multiple factors. This difficulty arises at two levels, i.e., the single-factor processing level and the multi-factor processing level. At the single-factor processing level, the semantic uncertainty and periodicity of a single factor should be appropriately expressed and effectively processed. The interaction between semantic uncertainty and periodicity should also be considered because it is a main source of difficulty in solving MPFP problems. At the multi-factor processing level, the interaction between multiple factors becomes a main source of difficulty in implementing effective data integration. The interaction is severely affected by the semantic uncertainty and periodicity in data of each single factor. An inappropriate representation and process of the data of a single factor will lead to incorrect selection and use of relevant knowledge about multiple factors; and in turn, lead to wrong solutions for the prediction problem at hand. Therefore, to represent and process the semantic uncertainty and periodicity simultaneously is an urgently-needed step to develop an effective solution for MPFP problems.

The literature review shows that existing techniques still have difficulty in providing effective solutions for MPFP problems for three main reasons: 1) many successful techniques have been reported for a single (fuzzy) time series; however, few are reported on processing a set of (fuzzy) time series at the same time; 2) semantic uncertainty processing is often limited to fixed semantic definition rather than changeable semantics; and periodicity processing is often based on a fixed-term period rather than flexible-term period; and 3) processing the semantic uncertainty and periodicity simultaneously has not yet been studied in any detail. Considering the gap between the great demand for applications with MPFP problems and existing techniques, developing an effective method for MPFP problems becomes a serious and important issue from both a theoretical and application perspective.

To resolve MPFP problems, this study applies complex fuzzy sets to represent the data of periodic factors. A complex fuzzy set (CFS) is an extension of a fuzzy set by replacing its real-valued membership function with a complex-valued membership function (Ramot *et al.*, 2002). Research shows that CFSs have potential applications in problems where "2-D membership function is used and there is significant interaction between these two dimensions of the membership (Dick, 2005)" such as problems in the social sciences (Nguyen *et al.*, 1998, 2000), engineering (Ramot *et al.*, 2003, 2002), and traffic congestion (Dick, 2005). In a CFS, the membership degree of any element u in a universe U is expressed by $r(u) \cdot e^{j\omega(u)}$, where $r(u)$ is called the modulus part and $\omega(u)$ is called the phase part. In the philosophy of CFS, the modulus part is used to describe a semantic feature and the phase part is for a temporal feature. A CFS can be introduced to solve MPFP problems as a representation model for data with the semantic uncertainty and periodicity. Furthermore, the operations on CFSs can be seen as processing models of multiple periodically changing factors under the above representation model. These features of CFSs, therefore, provide an operational way to solve MPFP problems.

Theoretically, CFSs provide potential for resolving the aforementioned MPFP problems. Based on this idea, this chapter presents a prediction method for the MPFP problems. First, the used data (observations or predictions) of the periodic factors in an MPFP problem are represented by CFSs. An aggregation operator, called the product-sum aggregation operator (PSAO), is then presented to integrate data of those factors, and its properties are discussed. The PSAO is different from existing aggregation operators because it integrates inputs distributed in a complex plane rather than a real interval. Finally, a PSAO based prediction method, the PSAOP method, for MPFP problems is proposed. Detailed steps of the method are illustrated. The main contribution of the study includes: 1) a method to resolve the MPFP problem, based on CFSs. This method deals with multiple periodic factors at the same time, and handles the semantic uncertainty and periodicity in the data used at the same time; and 2) it presents an innovative aggregation operator. The inputs of this aggregation operator are elements in the complex unit disc which are ordered in a quasi-order.

6.2 A product-sum aggregation operator

In this section, the product-sum aggregation operator (PSAO) is presented. This aggregation operator integrates data of multiple periodically changing factors in an MPFP problem. Section 6.2.1 introduces the basic concept of a CFS and two representation methods for the complex-valued membership degree. Section 6.2.2 presents a scale product operation for a CFS. Based on the scale product operation, Section 6.2.3 defines the constraint vector-sum (CVS) operation on CFSs. Extending the CVS and the scale product operations, the PSAO is presented in Section 6.2.4; and its relationship with ordinary product of vectors is also discussed.

6.2.1 Complex fuzzy sets

This section reviews some notations and operations on CFSs as the basis for further discussion. More details of the CFS operations and their potential applications can be found in Ramot *et al.* (2002). For convenience, let $\mathbb{R}$ and $\mathbb{C}$ be the sets of real and complex numbers, respectively; and $\mathbb{R}_{[0,1]}$ and $\mathbb{C}_{[0,1]}$ the real unit interval and the complex unit disc, respectively, i.e., $\mathbb{R}_{[0,1]} = \{a \in \mathbb{R} : 0 \leqslant a \leqslant 1\}$, and $\mathbb{C}_{[0,1]} = \{c \in$

$\mathbb{C} : |c| \leqslant 1\}$. Following general conventions, a complex number z will be denoted by $x + j \cdot y$ under the Descartes coordinate and by $r \cdot e^{j \cdot \omega}$ under the polar coordinate. The relation between these two expressions is $x = r \cdot \cos(\omega)$ and $y = r \cdot \sin(\omega)$. In this chapter, both expressions will be used according to the context of discussion.

Let U be a universe of discourse. A CFS $\tilde{A}$ on U is defined by a complex-valued membership function $\mu_{\tilde{A}}$ such that $\mu_{\tilde{A}} : U \to \mathbb{C}_{[0,1]}$, $u \mapsto r(u) \cdot e^{j \cdot \omega(u)}$, where $r(u)$ is a real-valued mapping from U to $\mathbb{R}_{[0,1]}$ and $\omega(u)$ is a periodic function (Ramot *et al.*, 2002). In the following, $r(u)$ is called the modulus part of $\mu_{\tilde{A}}(u)$; while $\omega(u)$ is called the phase part, accordingly. Under the Descartes coordinates, a CFS $\tilde{A}$ is therefore of the form

$$\tilde{A} = \left\{ \langle u, x(u) + j \cdot y(u) \rangle : |x(u)|, |y(u)| \in \mathbb{R}_{[0,1]}, x^2(u) + y^2(u) \leqslant 1 \right\}. \qquad (6.1)$$

6.2.2 A scale product on CFS

The introduced phase part in the complex-valued membership degrees is the main reason for difficulty of defining operations on CFSs. Except for the set-theoretical operations, Ramot *et al.* (2002) defined the set rotation and set reflection to handle the phase part from the viewpoint of geometrics. By restricting the periodic function ω with a fixed period 2π, Zhang *et al.* (2009) presented some concrete forms of set-theoretical operations that can also deal with the phase part. To resolve an MPFP problem, this study first presents the scale product of a CFS with respect to a complex number, which is an extension of the scale product on conventional fuzzy sets.

Definition 6.2.1. *Let $\tilde{A}$ be a CFS on U and $c \in \mathbb{C}_{[0,1]}$. The scale product of $\tilde{A}$ with respect to c is a mapping f_c from U to $\mathbb{C}_{[0,1]}$ such that for any $u \in U$, $f_c(u) = c \cdot \tilde{A}(u)$, where "$\cdot$" is the ordinary product of complex numbers.*

The mapping f_c is obviously a CFS on U and a combination of the compression and rotation operations. When c is a real number, f_c is a compression of $\tilde{A}$ in the sense that $|f_c| \leqslant |\tilde{A}|$. When c is a complex number and $|c| = 1$, then f_c is a rotation of $\tilde{A}$. In the following, we use $c \odot \tilde{A}$ to denote the scale product of a complex number c and a CFS $\tilde{A}$.

6.2.3 Constraint vector-sum of CFSs

The addition of complex numbers has special applications in engineering, which illustrates an integrated effect of a group of forces on an object. It can be extended to a set of CFSs, as shown below.

Suppose $\tilde{A}_1$, $\tilde{A}_2$, . . ., $\tilde{A}_n$ is a set of CFSs on U. The sum of these CFSs is a mapping s from U to $\mathbb{C}$, such that for any $u \in U$,

$$s(u) = \sum_{i=1}^{n} \tilde{A}_i(u) = \left(\sum_{i=1}^{n} x_i(u)\right) + j \cdot \left(\sum_{i=1}^{n} y_i(u)\right). \tag{6.2}$$

In general, $s(u)$ may not be a complex number in $\mathbb{C}_{[0,1]}$ so that it may not be a CFS. To make s a CFS, restriction on the modulus of each $s(u)$ is exerted. Therefore, the constraint vector-sum is defined below.

Definition 6.2.2. *Let $\tilde{A}_1$, $\tilde{A}_2$, . . ., $\tilde{A}_n$ be a set of CFSs on U. The constraint vector-sum (CVS) of $\tilde{A}_1$, $\tilde{A}_2$, . . ., $\tilde{A}_n$ is a mapping v from U to $\mathbb{C}_{[0,1]}$ such that for any $u \in U$*

$$v(u) = \min\left\{1, \sqrt{\left(\sum_{i=1}^{n} x_i(u)\right)^2 + \left(\sum_{i=1}^{n} y_i(u)\right)^2}\right\} \cdot e^{j \cdot \arctan\left(\frac{\sum_{i=1}^{n} y_i(u)}{\sum_{i=1}^{n} x_i(u)}\right)}, \tag{6.3}$$

where $\arctan\left(\frac{\sum_{i=1}^{n} y_i(u)}{\sum_{i=1}^{n} x_i(u)}\right)$ *is the principal argument of the phase. In the following, the right side of Eq. (6.3) will be denoted by $\bigoplus_{i=1}^{n} \tilde{A}_i(u)$ for convenience.*

Because $|v(u)| \leqslant 1$ for any $u \in U$, the mapping v is really a CFS over U. By

Definition 6.2.2, we need to emphasize that "the CVS of a set of n CFSs is defined with respect to the parameter n". In other words, the CVS is a one-off operation, which must be recalculated when the number of the CFSs involved varies. This characteristic is known because the ordinary associative law does not hold with respect to the CVS, i.e.,

$$\bigoplus_{i=1}^{n} \tilde{A}_i(u) \neq \left(\bigoplus_{i=1}^{n-1} \tilde{A}_i(u) \right) \oplus \tilde{A}_n(u).$$

Because the associative law is a powerful facility in many computation circumstances, if it holds with respect to the CVS, the computation can be simplified. Under some conditions, the associate law does hold with respect to the CVS.

Theorem 6.2.1. *Let $\tilde{A}_1$, $\tilde{A}_2$, ..., $\tilde{A}_n$ be a set of CFSs on U, if*

$$\sum_{i=1}^{n} |\mu_{\tilde{A}_i}(u)| \leqslant 1 \tag{6.4}$$

for any $u \in U$, the associative law holds with respect to the CVS of $\tilde{A}_1$, $\tilde{A}_2$, ..., $\tilde{A}_n$.

Proof. By Definition 6.2.2, it is only necessary to prove that

$$\sqrt{\left(\sum_{i=1}^{n} x_i(u) \right)^2 + \left(\sum_{i=1}^{n} y_i(u) \right)^2} \leqslant 1. \tag{6.5}$$

Because $\sum_{i=1}^{n} |\mu_{\tilde{A}_i}(u)| \leqslant 1$, we have

$$
\begin{aligned}
1 &\geqslant \left(\sum_{i=1}^{n} |\mu_{\tilde{A}_i}(u)|\right)^2 \\
&= \sum_{i=1}^{n}\left(\sqrt{x_i^2+y_i^2}\right)^2 + \sum_{i,j=1,i\neq j}^{n} 2\sqrt{x_i^2+y_i^2}\sqrt{x_j^2+y_j^2} \\
&\geqslant \sum_{i=1}^{n}\left(\sqrt{x_i^2+y_i^2}\right)^2 + \sum_{i,j=1,i\neq j}^{n} 2\sqrt{(x_ix_j)^2+(y_iy_j)^2+2x_ix_jy_iy_j} \\
&= \sum_{i=1}^{n}\left(\sqrt{x_i^2+y_i^2}\right)^2 + \sum_{i,j=1,i\neq j}^{n} 2\sqrt{(x_ix_j+y_iy_j)^2} \\
&= \sum_{i=1}^{n} x_i^2 + 2\sum_{i,j=1;i\neq j}^{n} x_ix_j + \sum_{i=1}^{n} y_i^2 + 2\sum_{i,j=1;i\neq j}^{n} y_iy_j \\
&= \left(\sum_{i=1}^{n} x_i\right)^2 + \left(\sum_{i=1}^{n} y_i\right)^2
\end{aligned}
$$

Hence, Eq. (6.5) holds. □

Furthermore, another weaker condition can be obtained, as shown below.

Theorem 6.2.2. *Let $\tilde{A}_1$, $\tilde{A}_2$, ..., $\tilde{A}_n$ be a set of CFSs on a universe U, if*

$$|\sum_{i=1}^{n} \mu_{\tilde{A}_i}(u)| \leqslant 1$$

for any $u \in U$, the associative law holds with respect to the CVS of $\tilde{A}_1$, $\tilde{A}_2$, ..., $\tilde{A}_n$. ■

Comparing the definitions of the CVS and the sum s in Eq. (6.2), the following inequality is obtained

$$\left|\bigoplus_{i=1}^{n} \tilde{A}_i(u)\right| \leqslant \left|\sum_{i=1}^{n} \tilde{A}_i(u)\right|.$$

6.2.4 A product-sum aggregation operator

The weighted-sum aggregation operation, i.e., $\sum_{i=1}^{n} w_i a_i$, where $\sum_{i=1}^{n} w_i = 1$ and $w_i \in \mathbb{R}_{[0,1]}$, is widely used in data fusion research. This operation can be extended to

CFSs by replacing each a_i by a CFS $\tilde{A}_i$, w_i by a complex number in $\mathbb{C}_{[0,1]}$, and the ordinary product by a scale product. We call the obtained operation the product-sum (PS) of a set of CFSs with respect to a complex-valued vector.

Definition 6.2.3. *Let $\tilde{A}_1$, $\tilde{A}_2$, ..., $\tilde{A}_n$ be a set of CFSs on U; $\boldsymbol{c} = (c_1, c_2, \ldots, c_n)$ a complex-valued vector in $\mathbb{C}^n_{[0,1]}$. The product-sum of $\tilde{A}_1$, $\tilde{A}_2$, ..., $\tilde{A}_n$ with respect to $\boldsymbol{c}$ is a mapping p_c from U to $\mathbb{C}_{[0,1]}$ such that for any $u \in U$, $p_{\boldsymbol{c}}(u) = \bigoplus_{i=1}^{n}(c_i \odot \tilde{A}_i(u))$.*

Obviously, the CVS is a special case of the product-sum when the complex-valued vector $\boldsymbol{c}$ is $(1, \cdots, 1)$. Another special case of the product-sum occurs when $\sum_{i=1}^n |c_i| \leqslant 1$. Under this condition, a product-sum is calculated by ordinary addition and multiplication of complex numbers.

Theorem 6.2.3. *Let $\tilde{A}_1$, $\tilde{A}_2$, ..., $\tilde{A}_n$ be a set of CFSs on U; $\boldsymbol{c} = (c_1, c_2, \ldots, c_n)$ a complex vector in $\mathbb{C}^n_{[0,1]}$. The product-sum of $\tilde{A}_1$, $\tilde{A}_2$, ..., $\tilde{A}_n$ with respect to $\boldsymbol{c}$ can be rewritten as: for any $u \in U$,*

$$p_{\boldsymbol{c}}(u) = \sum_{i=1}^{n}(c_i \odot \tilde{A}_i(u))$$

when $\sum_{i=1}^n |c_i| \leqslant 1$.

Proof. By Definition 6.2.2, it is sufficient to prove that the modulus of $\sum_{i=1}^n (c_i \odot \tilde{A}_i(u))$ is less or equal to 1 for any $u \in U$:

$$\begin{aligned}\left|\sum_{i=1}^{n}(c_i \odot \mu_{\tilde{A}_i}(u))\right| &\leqslant \sum_{i=1}^{n}\left|c_i \odot \mu_{\tilde{A}_i}(u)\right| \\ &= \sum_{i=1}^{n}|c_i| \odot |\mu_{\tilde{A}_i}(u)| \leqslant \sum_{i=1}^{n}|c_i| \leqslant 1\end{aligned}$$

□

Ramot *et al.* (2003) discussed a vector aggregation where $\sum_{i=1}^n |c_i| = 1$; hence, the vector aggregation is a special case of the product-sum. Keeping this fact in mind, it is natural to assume that the product-sum may be used as a new aggregation operator for uncertain information described by CFSs. This assumption will be checked below.

Given a set of CFSs, their product-sum is still a CFS. Hence, the product-sum is closed with respect to the operation and inputs.

Continuity is an important property of continuous aggregation operators, which means that minor modification of inputs will not result in a big change in the aggregation result. For a product-sum, we can check its continuity as follows. Suppose the complex-valued vector $\boldsymbol{c}$ is $(c_1, \ldots, c_n)$ and $c_i = x'_i + jy'_i$. For any $u \in U$, $p_c(u)$ is a complex number. Its modulus is given by Eq. (6.6).

$$r(p_c(u)) = \min\left\{1, \sqrt{\left(\sum_{i=1}^{n}(x_i(u)x'_i - y_i(u)y'_i)\right)^2 + \left(\sum_{i=1}^{n}(x_i(u)y'_i + y_i(u)x'_i)\right)^2}\right\}. \tag{6.6}$$

This is a continuous function with respect to $\mu_{\tilde{A}_i}(u) = x_i(u) + j \cdot y_i(u)$, $i = 1, 2, \ldots, n$. The principal argument of the phase of $p_c(u)$ is given by

$$\arg(v(u)) = \arctan\left(\frac{\sum_{i=1}^{n}(x_i(u)y'_i + y_i(u)x'_i)}{\sum_{i=1}^{n}(x_i(u)x'_i - y_i(u)y'_i)}\right). \tag{6.7}$$

This is a continuous function with respect to $\mu_{\tilde{A}_i}(u) = x_i(u) + j \cdot y_i(u)$, $i = 1, 2, \ldots, n$, except where $\sum_{i=1}^{n}(x_i(u)x'_i - y_i(u)y'_i) = 0$. Therefore, we arrive at the following conclusion.

Proposition 6.2.1. *The product-sum of a set of CFSs is continuous with respect to the modulus for each input CFS.* ■

Idempotence is another important property of most aggregation operators, which means that, if the inputs take the same value, the output after aggregating should be the same, i.e., generally given an aggregation operator Agg from $\mathbb{R}^n_{[0,1]}$ to $\mathbb{R}_{[0,1]}$ and $a \in \mathbb{R}_{[0,1]}$, $a = \text{Agg}(a, \ldots, a)$. This is called "strong idempotence" compared to the "weak idempotence" defined in Marichal (1998), where $a = \text{Agg}(a, \ldots, a)$ if and only if $a = 0, 1$.

Now we will check whether the product-sum of a set of CFSs satisfies "strong idempotence" or "weak idempotence". First, we will consider a special case, i.e., their CVS. Suppose $\tilde{A}_1 = \tilde{A}_2 = \cdots = \tilde{A}_n = \tilde{A}$ is a set of CFSs on a universe of discourse U and $u \in U$. Then the CVS of $\tilde{A}_1$, $\tilde{A}_2$, ..., $\tilde{A}_n$ is given by $v(u) = \min\{1, n \cdot r_{\tilde{A}}(u)\} \cdot e^{j \cdot \arg(\omega_A(u))}$. Suppose the idempotence of the CVS holds, then $r_{\tilde{A}}(u) \cdot e^{j \cdot \arg(\omega_A(u))} = \min\{1, n \cdot r_{\tilde{A}}(u)\} \cdot e^{j \cdot \arg(\omega_A(u))}$. Hence, $r_{\tilde{A}}(u) = 1$ or $r_{\tilde{A}}(u) = n \cdot r_{\tilde{A}}(u)$. So we get $\mu_{\tilde{A}}(u) = e^{j \cdot \omega_A(u)}$ or $r_{\tilde{A}}(u) = 0$, i.e., $\mu_{\tilde{A}}(u) = 0$. It seems that the CVS does not satisfy the "strong idempotence", but does satisfy the "weak idempotence".

Now, we will check the idempotence of the product-sum of a set of CFSs. Without loss of generality, set the complex-valued vector as $\boldsymbol{c} = (c_1, \ldots, c_n)$ and $c_i = x'_i + j \cdot y'_i$. Suppose $\tilde{A}_1 = \tilde{A}_2 = \cdots = \tilde{A}_n = \tilde{A}$ is a set of CFSs on a universe of discourse U and $u \in U$. Then the modulus of the product-sum $p_c(u)$ for $u \in U$ is shown in Eq. (6.8).

$$
\begin{aligned}
r(p_c(u)) &= \min\left\{1, \sqrt{\left(\sum_{i=1}^{n}(x(u)x'_i - y(u)y'_i)\right)^2 + \left(\sum_{i=1}^{n}(x(u)y'_i + y(u)x'_i)\right)^2}\right\} \\
&= \min\left\{1, \sqrt{(x^2+y^2)\left(\left(\sum_{i=1}^{n} x'_i\right)^2 + \left(\sum_{i=1}^{n} y'_i\right)^2\right)}\right\} \\
&= \min\left\{1, r_{\tilde{A}}(u) \cdot |\sum_{i=1}^{n} c_i|\right\},
\end{aligned}
\tag{6.8}
$$

and the principal argument of its phase is $\omega(p_c(u)) = \arctan\left(\frac{x\sum_{i=1}^{n} y'_i + y\sum_{i=1}^{n} x'_i}{x\sum_{i=1}^{n} x'_i - y\sum_{i=1}^{n} y'_i}\right)$. If the product-sum satisfies the idempotence, there should be

$$
r_{\tilde{A}}(u) = 1 \tag{6.9}
$$

or

$$r_{\tilde{A}}(u) = r_{\tilde{A}}(u) \cdot |\sum_{i=1}^{n} c_i|, \tag{6.10}$$

and

$$\frac{x\sum_{i=1}^{n} y'_i + y\sum_{i=1}^{n} x'_i}{x\sum_{i=1}^{n} x'_i - y\sum_{i=1}^{n} y'_i} = \frac{y}{x}. \tag{6.11}$$

By Eq. (6.9) and Eq. (6.11), we have $\sum_{i=1}^{n} y'_i = 0$, and $\tilde{A}(u) = e^{j\cdot\omega_{\tilde{A}}(u)}$. By Eq. (6.10) and Eq. (6.11), we have $\sum_{i=1}^{n} y'_i = 0$, and $\tilde{A}(u) = 0$ if $|\sum_{i=1}^{n} c_i| \neq 1$. The condition $\sum_{i=1}^{n} y'_i = 0$ indicates the sum of the complex-valued components of $\boldsymbol{c}$ is a real number. Thus, the above discussion indicates that the idempotence of the product-sum is related to the complex-valued vector $\boldsymbol{c}$. Under some conditions the idempotence of the product-sum holds.

Proposition 6.2.2. *The product-sum of a set of CFSs with respect to a complex-valued vector $\boldsymbol{c} = (c_1, \ldots, c_n)$ may be idempotent when the sum of c_is is a real number.*

6.2.5 A quasi-ordering on $\mathbb{C}_{[0,1]}$

Because the weak idempotence of an aggregation has a closed relationship with the natural partial order "$\leqslant$" on $\mathbb{R}_{[0,1]}$ and 0, 1 are the boundaries of $\mathbb{R}_{[0,1]}$, to confirm the CVS satisfies the weak idempotence requires a partial ordering on $\mathbb{C}_{[0,1]}$. This section discusses two existing partial orderings on $\mathbb{C}_{[0,1]}$ and presents a new ordering, i.e., the ordering by direction (OBD).

On the complex plane, the traditional partial ordering defined by the modulus (OBM) of a complex number is given as

$$a \leqslant b \iff |a| \leqslant |b| \tag{6.12}$$

for any $a, b \in \mathbb{C}$. This partial order can be restricted to $\mathbb{C}_{[0,1]}$. Under this restriction, it is clear that both 0 and $e^{j\omega}$ are located on the boundary of $\mathbb{C}_{[0,1]}$. Hence, the CVS satisfies the weak idempotence under the traditional partial order by modulus, i.e., $\bigoplus_{i=1}^{n} 0 = 0$, and $\bigoplus_{i=1}^{n} e^{j\omega} = e^{j\omega}$.

Dick (2005) defined another partial ordering on $\mathbb{C}_{[0,1]}$ by: for any $a, b, c \in \mathbb{C}_{[0,1]}$,

$$a \cdot b = c \Rightarrow c \leqslant a \text{ and } c \leqslant b. \tag{6.13}$$

By this partial ordering, the minimum is $0+0j$ and the maximum is $1+0j$. Comparing Dick's partial ordering and the traditional partial ordering by modulus, we reach the conclusion that they are equivalent to each other on $\mathbb{C}_{[0,1]}$.

Theorem 6.2.4. *The partial orderings given in Eq. (6.12) and Eq. (6.13) are equivalent to each other.*

Proof. ($\Rightarrow$) Consider the following three cases: 1) Suppose $a, b \in \mathbb{C}_{[0,1]}$ are two non-zero complex numbers; and $a \leqslant b$ under the partial ordering given in Eq. (6.12). Hence we have $|b| \neq 0$ and $|a| \leqslant |b|$. This means $|a|/|b| \leqslant 1$; therefore $a/b \in \mathbb{C}_{[0,1]}$. So, we have $a = b \cdot \frac{a}{b}$. By Eq. (6.13), $a \leqslant b$ and $a \leqslant a/b$. 2) If $b = 0$, then obviously, $a = 0$. We get $a \leqslant b$ under partial ordering in Eq. (6.13). 3) If $a = 0$ and $|b| \neq 0$, then we also get $a \leqslant b$ under partial ordering in Eq. (6.13) because $a = 0 = 0 + 0j$ is the minimum of the complex unit disc under partial ordering in Eq. (6.13). Summarizing the three cases, $a \leqslant b$ under partial ordering in Eq. (6.12) is also $a \leqslant b$ under partial ordering in Eq. (6.13).

($\Leftarrow$) Consider the following two cases: 1) $a = 0$. Obviously, for any $b \neq 0$ and $b \in \mathbb{C}_{[0,1]}$, we have $a = b \cdot \frac{0}{b}$. Because $\frac{0}{b} = 0 \in \mathbb{C}_{[0,1]}$, by Eq. (6.13), $a \leqslant b$. Hence, we have $0 = |a| \leqslant |b|$ and $a \leqslant b$ by Eq. (6.12). 2) Suppose $a \neq 0$ and $a \in \mathbb{C}_{[0,1]}$. Then for any $b \neq 0$, we have $a = b \cdot \frac{a}{b}$. If $a \leqslant b$ under partial ordering in Eq. (6.13), we require $\frac{a}{b} \in \mathbb{C}_{[0,1]}$. This means $|a/b| \leqslant 1$, i.e. $|a| \leqslant |b|$. By the partial ordering in Eq. (6.12), we have $a \leqslant b$. □

Introducing the phase part in the membership function is the main difference between a complex fuzzy set and a conventional fuzzy set. The above-mentioned two partial orderings are essentially defined based on modulus only and the influence of the phase part is not clearly addressed. By taking the three effects (i.e. positive, negative, and unclear) of a complex number as illustrated in Nguyen *et al.* (1998) into account, this chapter defines another ordering "$\leqslant$" over the complex unit disc $\mathbb{C}_{[0,1]}$ as shown below. For any $a, b \in \mathbb{C}_{[0,1]}$, and suppose $a = r_a \cdot e^{j\omega_a}$, $b = r_b \cdot e^{j\omega_b}$. An ordering $\leqslant$ by direction (OBD) is defined as

$$a \leqslant b \iff r_a \cos(\omega_a) \leqslant r_b \cos(\omega_b). \tag{6.14}$$

It can be proved that the given order $\leqslant$ is not a partial order because it is reflexive and transitive, but not antisymmetric. By the given OBD $\leqslant$, the minimum and the maximum of the unit disc of the complex plane are the complex number $-1 + 0j$ and $1 + 0j$. Notice that the CVS of n times $-1 + 0j$ is still $-1 + 0j$ (similarly for $1 + 0j$), hence the CFS still satisfies the weak idempotence.

The above OBD can be extended to a totally order by adding an extra condition with respect to the phase part. For any a, $b \in \mathbb{C}_{[0,1]}$, and suppose $a = r_a \cdot e^{j \cdot \omega_a}$, $b = r_b \cdot e^{j \cdot \omega_b}$. We say $a \leqslant b$ when

(1) if $r_a \cos(\omega_a) < r_b \cos(\omega_b)$; or

(2) if $r_a \cos(\omega_a) = r_b \cos(\omega_b)$, then $\omega_a \geqslant \omega_b$.

6.3 A PSAO-based prediction method for MPFP problems

This section presents a PSAO-based prediction (PSAOP) method to resolve MPFP problems. Section 6.3.1 addresses a formal description of the MPFP problem. Section

6.3.2 first outlines the process framework of the PSAOP method and then presents the detailed steps through two illustrative examples.

6.3.1 A formal address of an MPFP problem

Based on the principal characteristics discussed in Section 6.1, an MPFP problem is formally described as follows.

An event E is predictable through a set of factors a_1, a_2, ..., a_n. Each factor a_i varies following an approximate periodic pattern P_i which is modelled by a prediction method M_i established in advance $i = 1, 2, \ldots, n$. U is a period and is divided into two parts, i.e., U_0 and U_1. For U_0, the real observations $o_E(x)$ of event E and $o_i(x)$ of factor a_i at the moment $x \in U_0$ are collected; and for U_1, the predictions $p_i(y)$ of factor a_i at the moment $y \in U_1$ are obtained by the prediction model M_i, $i = 1, \ldots, n$. Now, for any $y \in U_1$, the MPFP problem seeks to answer the question of what is the prediction $p_E(y)$ of event E at y. Figure 6.1 gives a graphic illustration of the MPFP problem.

6.3.2 The PSAOP method for MPFP the problems

The primary process of the presented PSAOP method includes two stages, i.e., PS labelling and label retrieval. In the PS labelling stage, the observation $o_E(x)$ of event E at each moment $x \in U_0$ is assigned to the PS of observations $o_i(x)$, $i = 1, \ldots, n$, of those factors as its label. In the label retrieval stage, the PS of predictions $p_i(y)$, $i = 1, \ldots, n$, of those factors at a checking point $y \in U_1$ is calculated; and the most appropriate label for it is selected from U_0. The retrieved label is treated as the prediction $p_E(y)$ of event E at y, i.e., $p_E(y)$ is the solution for the MPFP problem.

The main task of the first stage is labelling the PS of observations of the factors. To achieve it, the observations of each factor are represented by complex-valued membership degrees (CMDs) of a CFS constructed according to the knowledge of the cause-

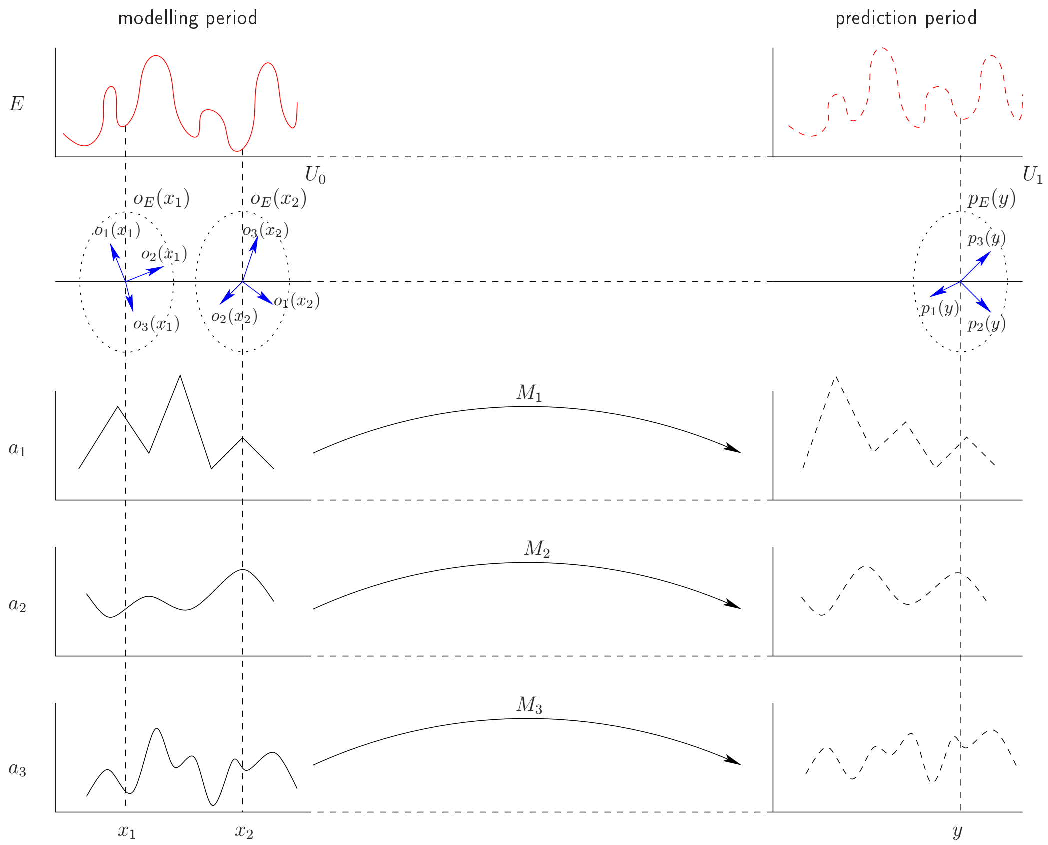


Figure 6.1: A graphic illustration of the MPFP problem. Suppose an event E is predicted by three periodic factors a_1, a_2, a_3. At different times, the impacts of the three factors are different as shown at times x_1 and x_2. Using the individual prediction methods M_1, M_2, and M_3, the predictions $p_1(y)$, $p_2(y)$ and $p_3(y)$ at time y can be obtained for the three factors, respectively. A solution of the MPFP problem gives the prediction $p_E(y)$ at y.

effect relationship between the factor and the event. At any moment $x \in U_0$, the PS of the corresponding CMDs indicates the integrated effect of those factors on event E at that moment; meanwhile, the real observation $o_E(x)$ of event E is another representation of the integrated effect of those factors. Therefore, we can label the PS with $o_E(x)$. After this process, the PSs at each moment $x \in U_0$ are labelled. Following the idea of PS labelling, the next task is to find a suitable label for the obtained PS of predictions of those factors at moment $y \in U_1$. This is conducted in the second stage. First, a prediction $p_i(y)$ of factor a_i is obtained through the prediction model M_i. The PS $\hat{p}_E(y)$ of the predictions $p_i(y)$, $i = 1, 2, \ldots, n$, can then be calculated. Next, similar

PSs to $\hat{p}_E(y)$ over U_0 are retrieved. Finally, a suitable label is selected from the labels of those PSs, which will be assigned to $p_E(y)$. This label is then treated as the solution of the PSAOP method. The detailed steps of the PSAOP method are listed below.

Stage 1: PS labelling

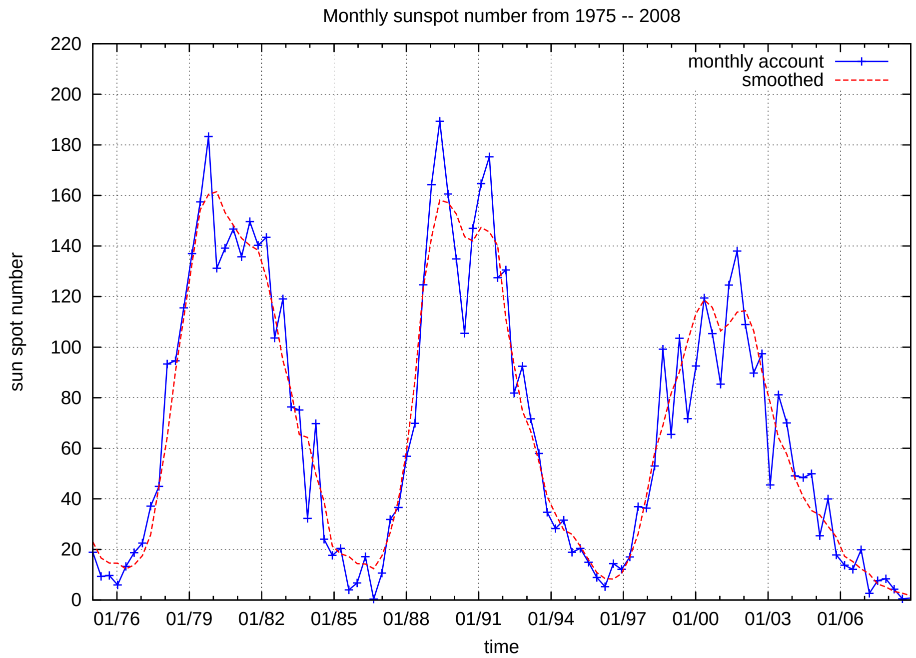
Step 1 Identify the approximate periods of the factors. Because the accurate period of a factor may not be exactly identifiable, an approximate period is used. As an example, let us consider the observations of monthly sunspot numbers from 1975 to 2008 (Figure 6.2(a)). An 11-year period has been chosen for this study. Therefore, the entire observations between Jan. 1976 and Jan. 2008 are approximately distributed in three cycles, i.e., Jan. 1976– Aug. 1986, Aug. 1986–Feb. 1996, and Feb. 1996–Jan. 2008.

Step 2 Identify the phase of each moment. Suppose T is the identified period of a factor a, $a \in \{a_1, \dots, a_n\}$. For any $u \in U_0$, u is assigned a value to indicate its phase with respect to the period T. Generally, we use 2π to match the period T and then set phase to each moment. For example, by observing the three cycles in Figure 6.2(a), it is known that the sunspot number approximately reaches its peak at the fourth year in each cycle. Therefore, a possible phase definition may set the phase of Jan. 1980 to be 0; the phases of Jan. 1976 and Aug. 1986 to be π and $-\pi$ respectively; and the others to be evenly displaced between $-\pi$ and π, i.e., the phase of Jan. 1979 is $\pi/4$ and the phase of Jan. 1982 is $-\pi/3$, etc.

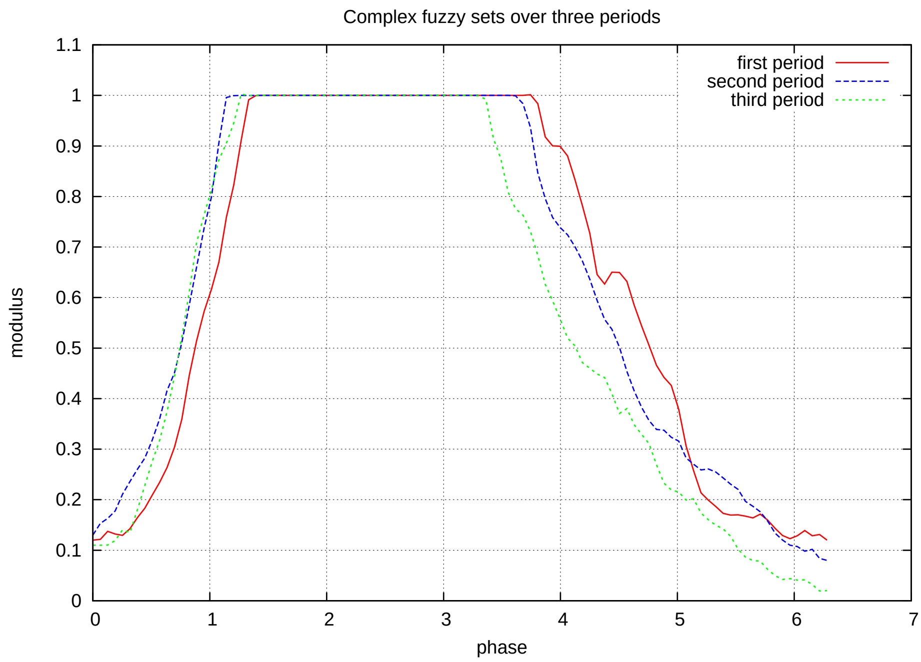
Remark. The range of phases need not be limited to 2π. We use 2π because it is easy to illustrate a phase in a simple and understandable way.

Remark. A definition of phase may not be restricted to evenly dividing a period by all time points. Sometimes, unevenly displacing phase values may be an appropriate choice.

Step 3 Determine the CFSs for those factors. The CFSs are extracted from accumulated knowledge about the cause-effect relationship between the factors and the



(a) Three cycles of the monthly sunspot number from 1975 to 2008 in an approximate 11-year period. (source: http://www.sidc.be)



(b) Modulus and phase for the three cycles.

Figure 6.2: Example of identifying the period and CFS of sunspot numbers

event. Let us return to the sunspot number example. A piece of knowledge about sunspot number and solar activity may be expressed by the statement that "a larger number of sunspots means stronger solar activity". Based on this knowledge, we will define a CFS for "larger sunspot number" in the following way: we say a sunspot number is "larger" if its smoothed value is larger than the 63.7 percents of the highest smoothed value in its cycle. Therefore, for an observation $o(x)$ of sunspot number at the moment x, we assign a CMD to it. The modulus part of the CMD is the degree of the sunspot number being a "larger sunspot number" and the phase part of the CMD is the phase of x with respect to the cycle. In this way, we obtain a definition of CFS which is roughly shown in Figure 6.2(b). We disclose some common features over the three periods: 1) the modulus value equals 1.0 when the phase value approximately falls in the interval $[\pi/2, 5\pi/4]$; 2) the modulus value increases sharply with the phase value when the phase value is in $[0, \pi/2]$; and 3) the modulus value decreases slowly with the phase value when the phase value is in $[5\pi/4, 2\pi]$. Based on these features, we refine the CFS for "larger sunspot number" as:

$$r(\omega) = \begin{cases} \dfrac{\omega}{\pi/2}, & \omega \in [0, \pi/2] \\ 1.0, & \omega \in (\pi/2, 5\pi/4) \\ \dfrac{\omega - 2\pi}{5\pi/4 - 2\pi}, & \omega \in [5\pi/4, 2\pi]. \end{cases} \tag{6.15}$$

Table 6.1 describes the complex-value membership degrees of some samples in the first period following this CFS.

Remark. The example here illustrates a relatively simple procedure of defining a CFS. In a real application, how to define a CFS is still an unsolved issue.

Step 4 Label the PSs of the observations of those factors. At each moment $x \in U_0$, the PS $C(x)$ of the CMDs for those factors is calculated by Eq. (6.3). The observation $o_E(x)$ of event E is assigned to $C(x)$ as its label.

Stage 2: Label retrieval

Step 5 Generate the predictions of those factors at the checking point y in U_1.

Table 6.1: Complex-valued membership degrees of samples in 1976–1986

Year-Month	real observation	smoothed value	modulus	phase (π)
1976-06	12.2	12.2	0	0
1978-06	95.1	89.3	0.78	0.39
1979-06	149.5	153	1	0.59
1980-06	157.3	154.7	1	0.78
1981-06	90.9	141.5	1	0.97
1982-06	110.4	117.3	1	1.17
1983-06	91.1	70.5	0.85	1.37
1984-06	46.1	46.5	0.59	1.56
1985-06	24.2	18	0.32	1.76
1986-06	1.1	13.8	0.07	1.95
1986-09	3.8	12.3	0	2

The prediction $p_i(y)$ of each individual factor a_i is obtained by its prediction model M_i, which is established in advance.

Step 6 Compute the PS of those predictions at the checking point y. This step is straightforward. The main purpose of this step is to integrate those predictions to obtain the PS $C_E(y)$ at y.

Step 7 Search the most similar PSs to $C_E(y)$ **from** U_0. For the predicted PS $C_E(y)$ at the checking point y, its similar PSs in U_0 are retrieved by the following similarity measurement:

$$C_E(x) = C_E(y) \text{ iff } |C_E(x) - C_E(y)| < \epsilon, x \in U_0, \tag{6.16}$$

where ϵ is a given error. Suppose $x_1, x_2, \ldots, x_m$ is a set of moments in U_0 such that

$$C_E(x_1) = C_E(x_2) = \cdots = C_E(x_m) = C_E(y), \tag{6.17}$$

and $o_E(x_1)$, $o_E(x_2)$, ..., $o_E(x_m)$ are their labels.

Step 8 Generate prediction $p_E(y)$ **of event** E **at the checking point** y. This step will select one label from $o_E(x_1)$, $o_E(x_2)$, ..., $o_E(x_m)$ as the prediction $p_E(y)$.

Noting that the prediction $P_E(y)$ is affected by the module and the phase parts of the CFSs used, we can identify three possible selection standards, i.e., on them both, on module parts only, and on phase parts only, in a real application. In this study, we use a three-level-nearest-set selection method to combine the three standards.

Initially, the first-level nearest set is selected based on a threshold δ and the generalized distance between y and x_k ($k = 1, \ldots, m$) measured by Eq. (6.18), where $\arg(\omega)$ is the principal argument of phase ω and $ran(T_i)$ is the range of the phase values. [1] Suppose $X(\delta)$ is the selected set and $O(\delta)$ is the set of labels.

$$d(y, x_k) = \max \Big\{ |r_i(y) - r_i(x_k)|, \frac{1}{ran(T_i)} \left| \arg(\omega_i(y)) - \arg(\omega_i(x_k)) \right|, i = 1, 2, \ldots, n \Big\} \leqslant \delta, \tag{6.18}$$

Secondly, the second-level nearest set is constructed on top of the first-level nearest set by setting another threshold ω about phases such that

$$\max\{|\omega_i(x) - \omega_i(y)|, i = 1, 2, \ldots, n\} \leqslant \omega. \tag{6.19}$$

Suppose the determined second-level nearest set is $X(\delta, \omega)$.

Thirdly, on top of the second-level nearest set, a third-level nearest set is obtained by setting a third threshold γ for modulus such that

$$\max\{|o_i(x) - p_i(y)|, i = 1, 2, \ldots, n\} \leqslant \gamma. \tag{6.20}$$

Suppose the obtained third-level nearest set is $X(\delta, \omega, \gamma)$; accordingly, the label set is $O(\delta, \omega, \gamma)$.

Finally, select the most occurring label(s) in $O(\delta, \omega, \gamma)$ as the prediction $p_E(y)$.

[1] This used distance is borrowed and extended from that in Zhang *et al.* (2009).

Below is an example to illustrate this selection procedure. Suppose at four moments, i.e., x_1, x_2, x_3, and x_4, the condition Eq. (6.17) holds with respect to three factors as shown in Table 6.2, where C_i is the corresponding CFS of the factor a_i, $i = 1, 2, 3$. Then by Eq. (6.18), the generalized distance between x_k and y is calculated, which is shown in Table 6.3, $k = 1, 2, 3, 4$.

Table 6.2: Observations and predictions about three factors

	Prediction	Observations		
Element	C_E	C_1	C_2	C_3
x_1	$0.89e^{j0.24}$	$0.24e^{j0.15}$	$0.92e^{j0.62}$	$0.38e^{j0.32}$
x_2	$0.89e^{j0.24}$	$0.59e^{j0.69}$	$0.18e^{j0.45}$	$0.35e^{j2.34}$
x_3	$0.89e^{j0.24}$	$0.50e^{j0.94}$	$0.17e^{j0.80}$	$0.55e^{j2.17}$
x_4	$0.89e^{j0.24}$	$0.05e^{j0.51}$	$0.89e^{j0.61}$	$0.33e^{j2.86}$
y	$0.89e^{j0.24}$	$0.48e^{j0.6}$	$0.87e^{j0.09}$	$0.42e^{j1.24}$

Table 6.3: Example of distance measurements between y and xs (with 2 valid position)

factors	C_1		C_2		C_3	
	modulu r	phase ω	modulu r	phase ω	modulu r	phase ω
x_1	**0.24**	0.07	0.05	0.08	0.04	0.15
x_2	0.12	0.01	**0.69**	0.06	0.07	0.17
x_3	0.03	0.05	**0.71**	0.11	0.13	0.15
x_4	**0.43**	0.01	0.02	0.08	0.09	0.26

Suppose $\delta = 0.7$, $\omega = 0.20$, and $\gamma = 0.25$, then the first-level nearest set is $\{x_1, x_2, x_4\}$; the second-level nearest set is $\{x_1, x_2\}$; and the third-level nearest set is $\{x_1\}$. Because the third-level nearest set contains the unique element x_1, the observation $o_E(x_1)$ is assigned as the prediction $p_E(y)$ of the event at time y.

Remark. (1) The three-level nearest set is not necessarily required in some applications where flexible constraints may be suitable for real settings. Accordingly, the three measure parameters δ, ω, γ for distances are also determined based on real settings. (2) Each level nearest set may contain labels with the same occurring number. In this situation, the selected label as prediction should be concerned with the demand

and nature of real applications. For instance, if the prediction is about potential natural disaster, an alert label may be better than a non-alert one.

6.4 Experiments using the PSAOP method

In this section, we apply the PSAOP method to two applications. The first experiment is to predict the annual sunspot number; and the other experiment is to predict the daily fire danger rating in a fire season in Sydney. We choose these two applications because they have typical characteristics of MPFP problems as discussed above. To conduct these experiments, the presented PSAOP method is programmed in C++ programming language with the support of MySQL database on a Fedora Linux distribution and runs on a Dell Laptop with a 2.8GHz Intel Core II Duo processor and 4 GB RAM. Below are the main steps and results.

6.4.1 Annual sunspot number prediction

6.4.1.1 Experiment description

Sunspot number is an important indicator of solar activity. It is commonly recognized that the solar activity follows a 11-year cycle which is observed from the annual sunspot numbers. Our first experiment uses the annual sunspot dataset downloaded from the Solar Influences Data Analysis Centre (SIDC, http://www.sidc.be). Following the setting in Aghakhani and Dick (2010), we use the observations during years 1700–1920 (U_0) as the training data, and those during years 1920–1979 (U_1) as the testing data. Noting that the dataset is a single time series which means the prediction involves only one factor, the experiment adjusts slightly the steps listed in Section 6.3 to fit this data. The main steps of the experiment are illustrated below.

Stage 1: PS labelling.

From the dataset use, we identified the three most occurring periods, i.e., 9-year, 10-year, and 11-year cycles to construct three CFSs. Each CFS is defined as follows: let $T = [x_{start}, x_{end})$ be a given period, $n_{\max}$ and $n_{\min}$ be the maximum and minimum values of sunspot number in T, and $x_{\max}$ be the year with the maximum sunspot number in T, for each year x in T, its complex membership degree is given by

$$\mu_T(x) = \begin{cases} \dfrac{n(x) - n_{\min}}{n_{\max} - n_{\min}} \cdot e^{j\frac{\pi\cdot(x-x_{start})}{x_{\max}-x_{start}}}, & x_{start} \leqslant x \leqslant x_{\max} \\ \dfrac{n(x) - n_{\min}}{n_{\max} - n_{\min}} \cdot e^{j(\frac{\pi\cdot(x_{end}-x)}{x_{end}-x_{\max}}+\pi)}, & x_{\max} < x \leqslant x_{end} \end{cases}$$

Therefore, we have three CFSs, i.e., μ_9, μ_{10}, and μ_{11} and can calculate the PS at each year in U_0. For each PS, we use the real observation of the sunspot number at that year as its label.

Stage 2: Label retrieval.

Based on the three identified periods, we predict the CMDs of a year in U_1 as follows: we identify a year before 1921 as the common starting year of the three periods; and estimate the phases of each year in U_1 with respect to the three periods; moreover, we add a small random offset to the phase of the middle point of the period. After setting the phase value of a year y in U_1, we assign the average value of modulus of years in U_0, which have equal phase value to y, as its modulus. Thus, we obtain a prediction of CMD for year y. Then we can calculate the PSs of each year in U_1 and generate a prediction of the annual sunspot number following the steps in Section 6.3.

Figure 6.3 shows the prediction results compared with the real observation.

6.4.1.2 Comparison and analysis

To evaluate the performance of the PSAOP method on this dataset, we measure the error with three metrics used in Aghakhani and Dick (2010): mean square error

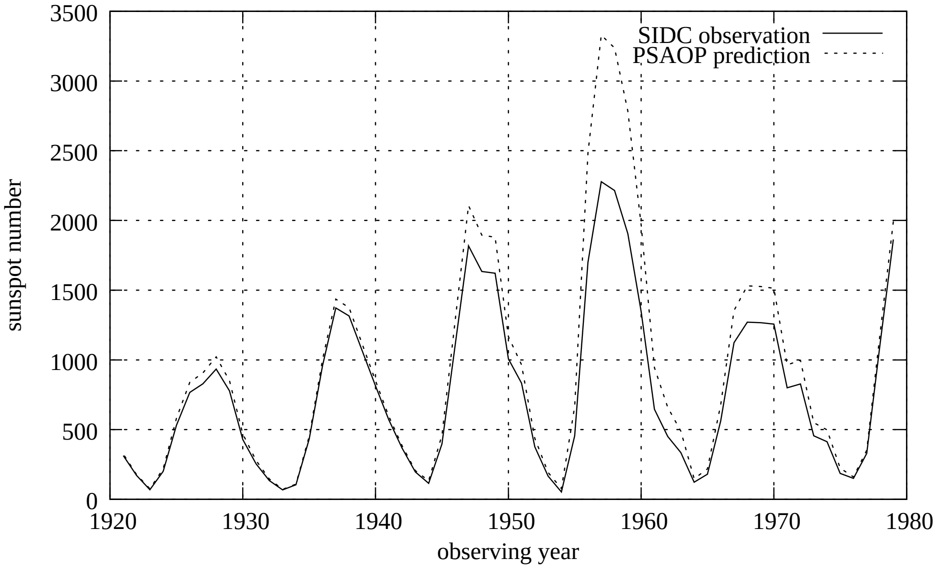


Figure 6.3: Real observation vs. PSAOP prediction for the sunspot dataset

(MSE), normalized MSE (NMSE), and non-dimensional error index (NDEI):

$$MSE = \sum_{i=1}^{n} \frac{(p_i - o_i)^2}{n} \tag{6.21}$$

$$NMSE = \frac{1}{\sigma^2 n} \sum_{i=1}^{n} (p_i - o_i)^2 \tag{6.22}$$

$$NDEI = \frac{MSE}{\sigma} \tag{6.23}$$

where o_i is the real observation and p_i is the prediction, and σ is the standard deviation of the dataset. Table 6.4 compares the performance of the PSAOP and the ANCFIS method in Aghakhani and Dick (2010).

Table 6.4: Error measurements comparison

Method	NMSE	MSE	NDEI
ANCFIS (online)	0.1473	9.2×10^{-3} (0.183)[2]	0.38
PSAOP (averaged)	0.249	0.323	0.499

Observing Figure 6.3 and compared with the error measurements in Aghakhani

[2]We noted that the MSEs of the two methods have a big gap. We think the reason for this gap is that the method used a different scale for representing the observations and predictions because the MSE of ANCFIS is about 0.183 by our calculation under a scale factor 1/500 (Graves and Pedrycz, 2009).

and Dick (2010), we noted that the PSAOP method is able to generate predictions to an acceptable extents for most instances using a simple setting for the CFSs. This indicates that the PSAOP method is a rational process method. However, we also noted exceptions for instances such as years between 1954–1964, the errors of the PSAOP method are over-estimates. Reasons for these exceptions may include: 1) The PSAOP method is initially designed for MPFP problems with multiple factors rather than a single factor; therefore, it has limitations in processing a single factor; 2) The CFSs used in the experiment are defined in a very simple way, which may not fully fit the specific features of the problem; and 3) The PSAOP method is a kind of declarative method whose ability of self-learning is weaker than inductively-learned methods such as the ANCFIS. Hence, more work is needed to amend the PSAOP method. Moreover, the results from the PSAOP method and by the ANCFIS method in Aghakhani and Dick (2010) indicate that CFS-based techniques are potential choices for such types of prediction problems.

6.4.2 Bushfire warning prediction

6.4.2.1 Brief background

Our second experiment is conducted to predict bushfire danger rating. Bushfire is one of main natural disasters in Australia, with has strong seasonality. Roughly speaking, a fire season in Australia runs from October to March with one or two months variation before or after, in different areas and years (Emergency Management Australia, 2010). Australia's primary referenced index of bushfire danger is the daily Fire Danger Index (FDI) provided by the Australia Bureau of Meteorology (ABM). The FDI is approximately equal to the Forest Fire Danger Index (FFDI) or the Grass Fire Danger Index (GFDI) (Lucas, 2010) and is calculated according to four primary meteorological indicators: "maximum temperature", "efficient precipitation", "wind speed", and "relative humidity" (Finkele *et al.*, 2006). Currently, a newly-released six-grade national fire danger rating system is adopted in Australia (see Table 6.5).

Table 6.5: Fire danger rating of the Australia Bureau of Meteorology

Category	Fire Danger Index (FDI)
Catastrophic (Code Red)	100+
Extreme	75–99
Severe	50–74
Very High[3]	25–49
High	12–24
Low-Moderate	0–11

Source: (http://www.bom.gov.au)

6.4.2.2 Experiment description

To conduct this study, the data, observed at the Sydney Airport climatological station, New South Wales, Australia from Jun. 01, 1972 to Dec. 31, 2010, is used. The dataset includes the daily observations of the aforementioned four typical meteorological indicators, i.e. "maximum temperature (T)", "rainfall (P)", "relative humidity (RH)", and "wind speed (W)", which will be used in this experiment as factors determining the FDI. Taking the entire dataset, we use the observations between Jan. 01, 2000 and Aug. 31, 2009 as the training data and those between Sept. 01, 2009 and Dec. 31, 2010 as the testing data. The detailed steps are illustrated below.

First, we formalize the problem as follows. Let E be the event that "find the FDI of a day" and the possible FDI of the day d be $E(d)$. To predict $E(d)$, four meteorologic indicators are used "maximum temperature" (a_T), "relative humidity" (a_H), "wind speed" (a_W), and "rainfall" (a_R). If the observations on these four indicators have been obtained for one day (y) in a fire season, what is the possible FDI for that day?

In the following, let $o_I(d)$ and $\omega_I(d)$ be the observation and phase of a day d with respect to indicator I when referring to a special indicator I.

Stage 1: PS labeling.

The main fire seasons in Sydney include Spring and Summer, i.e., from October to March. Considering possible seasonal variations, this study sets the fire season to run from September 01 to March 31. For illustrative purpose, this study uses the same 12-month period for the four indicators and maps the phases to $[0, 2\pi)$. Therefore, the phase value of each day in a calendar year is identified as shown in Figure 6.4. The figure is read: for example, the phase value of the day "January 01, 2001" for the indicator "temperature" is π; for the indicator "relative humidity" (or "rainfall") is 0.83π; and for the indicator "wind speed" is 0.67π.

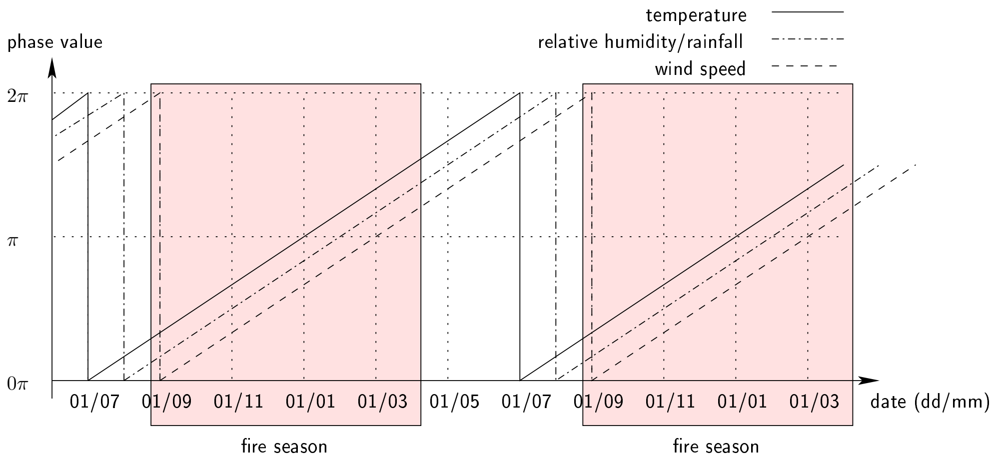


Figure 6.4: Phase value settings for the four meteorological indicators

This study sets four CFSs for typical "weather words" of the four meteorological indicators (http://www.bom.gov.au).

- temperature: A CFS "high temperature" is defined and denoted by C_T. For any day d between October 1 and March 31, let

$$C_T(d) = e^{j\omega_T(d)} \cdot \begin{cases} 1.0, & o_T(d) > 32.0 \\ \dfrac{o_T(d) - 16.0}{32.0 - 16.0}, & o_T(d) \in [16.0, 32.0] \\ 0.0, & o_T(d) < 16.0; \end{cases}$$

and for any day d between April 1 and September 30

$$C_T(d) = e^{j\omega_T(d)} \cdot \begin{cases} 1.0, & o_T(d) > 16.0 \\ \dfrac{o_T(d) - 10.0}{16.0 - 10.0}, & o_T(d) \in [10.0, 16.0] \\ 0.0, & o_T(d) < 10.0. \end{cases}$$

- relative humidity: A CFS "lower relative humidity" is defined and denoted by C_H:

$$C_H(d) = e^{j\omega_H(d)} \cdot \begin{cases} 1.0, & o_H(d) < 45.0 \\ \dfrac{o_H(d) - 70.0}{45.0 - 70.0}, & o_H(d) \in [45.0, 70.0] \\ 0.0, & o_H(d) > 70.0 \end{cases}$$

- wind speed: A CFS "high wind speed" is defined and denoted by C_W:

$$C_W(d) = e^{j\omega_W(d)} \cdot \begin{cases} 0.0, & o_W(d) < 10.0 \\ \dfrac{o_W(d) - 10.0}{30.0 - 10.0}, & o_W(d) \in [10.0, 30.0] \\ 1.0, & o_R(d) > 30.0 \end{cases}$$

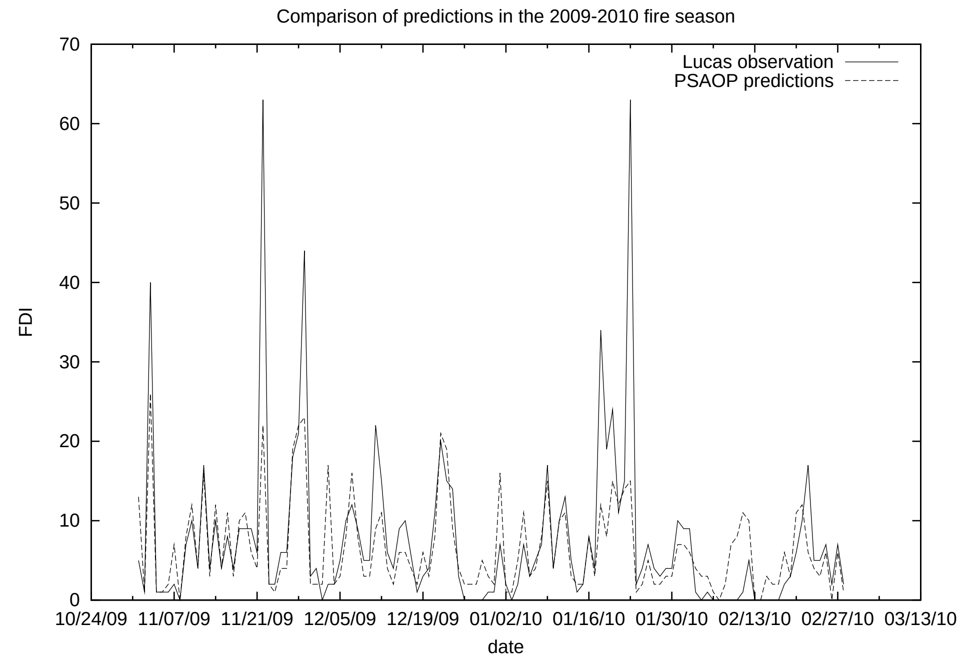
- rainfall: A CFS "lower rainfall" is defined and denoted by C_R:

$$C_R(d) = e^{j\omega_R(d)} \cdot \begin{cases} 1.0, & o_R(d) < 4.8 \\ \dfrac{o_R(d) - 48.0}{4.8 - 48.0}, & o_R(d) \in [4.8, 48.0] \\ 0.0, & o_R(d) > 48.0 \end{cases}$$

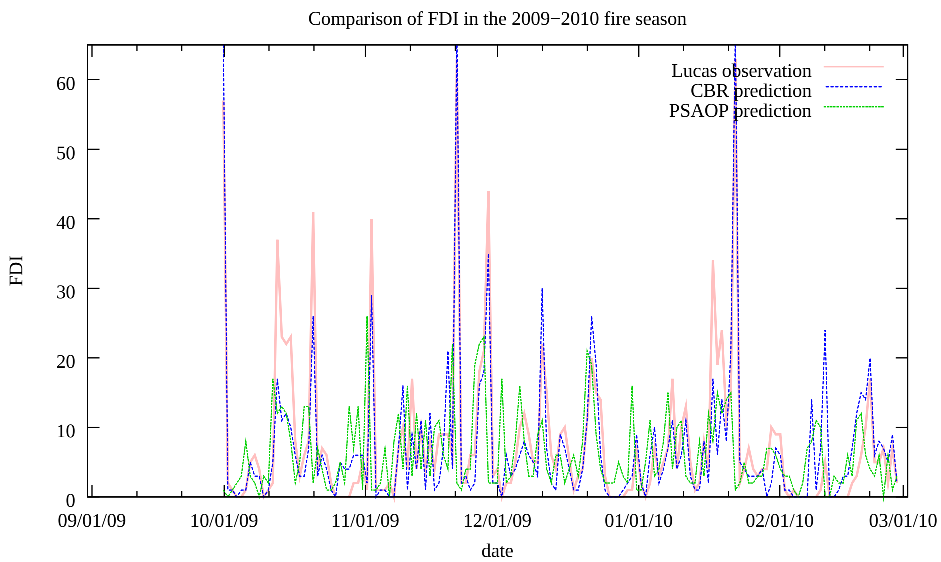
Based on the four CFSs, we can calculate the PS of each day for the training data and label it with the real fire danger rating at that day.

Stage 2: Label retrieval.

The PSs of the testing data are calculated according to the CFSs. Following the steps in Section 6.3, we obtain predictions for FDI. Figure 6.5(a) illustrates the PSAOP predictions of FDI and those in the original dataset.



(a) Prediction of the PSAOP.



(b) Comparison of the PSAOP and a CBR method.

Figure 6.5: Comparison of preditions by the PSAOP, a CBR method, and the original dataset. Note: the CBR method is developed for this experiment.

For comparison purposes, we developed a case-based reasoning (CBR) method following the basic CBR framework. The prediction result of the CBR method is compared with the PSAOP method as shown in Figure 6.5(b).

6.4.2.3 Comparison and analysis

To further evaluate the PSAOP method, we also developed another case-based reasoning (CBR) method using three commonly-used distance measurements (i.e., the Euclidean distance, the Manhattan distance, and the Chebychev distance) whose predictions are shown in Figure 6.5. We also conducted other experiments for the PSAOP method and the CBR method on the collected data by adjusting the weights of the factors used and changing the PS labelling and retrieval strategies. The performance of these experiments are summarised in Table 6.6. We measure the performance of the PSAOP and the CBR method using the error metric MSE, NMSE in Aghakhani and Dick (2010) and an reported fire danger rating accuracy:

$$FDRA = \frac{\text{number of correct fire danger rating}}{\text{number of total days}}. \tag{6.24}$$

Observing Figure 6.5 and Table 6.6, we conclude that: 1) the PSAOP method can correctly generate predictions which are consistent with the main trend of the FDI, although it has gaps with different CBR methods; 2) compared with the predictions from a CBR method, its prediction has similar accuracy in terms of fire danger rating; and 3) adjusting weights and changing PS labelling and retrieval strategies are possible ways to improve the performance. Moreover, we noted that the performance of a CBR method varies with respect to the selection of a distance measurement used in this dataset. Because these experiments are only conducted on this data in this study, further study is required.

We identify some reasons for the gap of FDI prediction. First, the meteorological indicators change so quickly that a simple CFS may not fit those changes, because little has been known about how to define a CFS appropriately since the CFS was

Table 6.6: Performance comparison of different amendments

Method with different strategy	MSE	NMSE	FDRA
CBR (Euclidean distance with equal weight to factors)	24.64	0.12	0.87
CBR (Euclidean distance with different weights to factors)	35.61	0.17	0.84
CBR (Chebychev distance)	99.09	0.49	0.85
CBR (Manhattan distance)	99.08	0.48	0.84
PSAOP (with equal weight to factors and uses algorithm mean to retrieve label)	47.22	0.23	0.74
PSAOP (with different weights to factors and uses most occurred as label)	64.13	0.31	0.79
PSAOP (with equal weight to factors and uses most occurred as label)	89.80	0.44	0.86
PSAOP (with different weights to factors and uses most occurred as label)	56.39	0.27	0.85

presented. The CFSs used here are defined only based on common knowledge. The prediction result, however, has high accuracy from the viewpoint of fire danger rating. This means CFS is potentially a powerful tool for such types of problems; but, more effort is needed to solidify its theoretical and practical basis.

Secondly, the introduction of phase is a unique feature of a CFS which is different from other information representations. The processing mechanism of a phase value has not been well discussed, interpreted, and established. In our case study, we set the approximate period of each indicator to be 12-months and the phase of each day is distributed evenly. This setting works well in the majority of situations. However, it may not be appropriate. Moreover, the phase values will affect the PS of a set of data and, in turn, affect the label retrieval. This also leads to the gap. Noting that meteorological conditions change rapidly and a fixed-length period has limitations with catching up with its change, we think a varied-length period may be more suitable for these kinds of applications.

Thirdly, processing semantic uncertainty and periodicity at the same time is a difficulty problem. The PSAOP provides a method to achieve that goal. It is by no means a perfect one, and there is more work required to amend it. For example, Lucas (2010) indicates that the "drought factor" is an important factor to determine the FDI. It is different from the indicator "rainfall" used in this experiment. We use "rainfall" in this experiment because it is related to "drought factor" and it is a standard indicator in daily meteorological observation so that people understand the definition of the CFS from their experience. The use of "rainfall" may also account for the errors.

An encouraging fact is that the above case studies show that the PSAOP method uses the advantages of CFS. For instance, it only uses some intuitive concepts such as "lower humidity", "high temperature", and "high wind speed" rather than specific measurements; and the given definitions of the CFSs are designed according to understandable knowledge; however, the prediction result is accurate. In particular, it can deal with uncertainty and periodicity simultaneously. This indicates that the PSAOP method is an applicable method for these types of problems.

6.5 Conclusions

This chapter presents a CFS-based predication method, the PSAOP method, for solving MPFP problems. In the PSAOP method, past experience and knowledge with respect to the factors and the predicted event are firstly formalized by CFSs; observations about those factors are then represented by complex-valued membership degrees. A novel product-sum aggregation operator, the PSAO, is used to integrate data from these factors; finally, the prediction is made by searching the label for the product-sum of current data. The PSAOP method is demonstrated with its application to annual sunspot number prediction and bushfire danger rating prediction. Our study indicates that: 1) the PSAOP method is simple but has high performance and adaptability. It takes advantage of the CFS to describe semantic uncertainty and periodicity in information; 2) the PSAO can deal with the semantic uncertainty and periodicity in data from multiple periodically changing factors in MPFP problems simultaneously. In particular, the PSAO can integrate inputs with a quasi-order.

During the case studies, we observed that there are still important issues to be studied. Firstly, the presented PSAOP method closely depends on the time series prediction methods for each factor. How to effectively use existing data (in particular historic records) to build CFSs is still little to known. This is an important theoretical and practical issue in CFS-related research. Secondly, when the observations of factors are qualitative expressions, the accuracy of some time series methods may be lost; hence, new methods for processing time series with qualitative data are required. We have noted relevant studies in the framework of fuzzy time series. These studies may be helpful for improving the performance of the presented PSAOP method. Thirdly, the phase part of each complex-valued membership degree is a unique feature of the CFS other than the conventional fuzzy sets. However, how to rationally define and interpret it is still a big question when applying CFSs to real applications. Although the used phase definition is applicable in our experiments, developing a rational method to define the phase part is still a pressing issue because it may affect the performance of the presented method. We will pay more attention to this issue. Finally, the presented PSAOP method is a declarative method which has remarkable difference to inductively-learning prediction methods such as the ANCFIS method. Although we noted that the performance can be improved by adjusting the weights of factors and selecting more appropriate labelling strategy, these issues are not detailed discussed in this chapter and will be studied further in the future.

Chapter 7 Decision Information Similarity Measurement

Group decision-making is a primary strategy used to develop decision support applications for use in counter-terrorism, government management, and business intelligence development. Too many similar opinions from participants can mislead the final decision. Measuring similarity between opinions of participants (MOSP) in advance is an important strategy to reduce this risk. Due to the lack of opinion data for a focal topic and the varieties of opinion representations, measuring the similarity is difficult and has not been well-studied in developing decision support. Noting that the similarities gradually change from different viewpoints, this chapter develops a gradual aggregation algorithm and establishes a method based on it, called the three-level-similarity measuring (TLSM) method, to measure the similarity at three similarity levels, i.e. the assessment level, the criterion level, and the option level. Two applications of the TLSM method on social policy selection and energy policy evaluation are conducted. The study indicates that the TLSM method can effectively measure the similarity between opinions with small-size, or possibly missing, and multiple-nature opinion data.

The rest of the chapter is organized as follows. Section 7.1 overviews the MOSP problem and the idea of the TLSM method. Section 7.2 develops a gradual aggregation algorithm which is used to generate an overall opinion similarity. In Section 7.3 we introduce the TLSM method in detail. Section 7.4 illustrates two case studies for applying the TLSM method to social policy selection and energy policy evaluation problems.

7.1 Introduction

Multiple-criteria group decision-making (MCGDM) is recognized as an efficient strategy in many organizational group decision problems (Kou *et al.*, 2011; Munda, 2008; Triantaphyllou, 2000), where a final decision is made based on participants' opinions on candidate options. If these participants' opinions are overly similar, a decision may be misleading. This increases the chance of putting an inappropriate decision into effect. Making an appropriate strategically decision in a large group is a time-consuming and costly task; however, tuning an inappropriate decision will cost even more. To reduce this risk, measuring opinion similarity between participants (MOSP) in advance is, therefore, an important issue in developing decision support for essential decision problems relating to such issues as counter-terrorism, business intelligence, nuclear inspection, government management, and others.

Opinion similarity has wide applications in various fields, such as online recommender systems (Adomavicius and Tuzhilin, 2005; Hakanen *et al.*, 2011; Symeonidis *et al.*, 2010; Vojnović *et al.*, 2009; Yu and Lai, 2011). However, the MOSP problem is still an unsolved and challenging issue (Mata *et al.*, 2009). Difficulties in solving the MOSP problem include the effective processing of small-size opinion data and of varied opinion representations. Due to the restrictions on funding, time, cost, private policies, and other reasons, a decision is often made by a limited number of participants. The total amount of usable opinions for measuring similarity is small sized, even though all participants would like to express their opinions thoroughly. The small-size opinion data makes it is very hard to apply methods for large-size data to the MOSP problem. Varied opinion representation is another difficulty in solving the MOSP problem. Participants prefer to express their opinions in their own ways based on their understandings of, and experiences in a given decision topic. However, this is bound to difficulties for measuring the similarity between their opinions. A strategy commonly used to regulate opinion representation is providing a fixed number of choices, for example, some predefined linguistic terms, or a set of ordinal numbers (Herrera *et al.*, 2001; Munda, 2008). However, this cannot completely avoid varied opinion representations because the pre-defined choices may have different semantics for different evaluation criteria. A third difficulty in solving the MOSP problem is the lack of a fixed reference point for the measuring task. A person's opinion is a type of subjective information, which varies from one object to another. Two participants may have similar opinions on some options, but completely different opinions on the others. Hence, the reference point should be able to fit these changes. The MOSP problem needs to find the similarity between two participants on a decision problem as a whole; so the reference point should be used at different levels.

Keeping the aforementioned difficulties in mind, this chapter presents a measuring method to solve the MOSP problem. The method is based on three assumptions: 1) Given a criterion, if two participants' opinions are similar for the majority of testing, it is rational to presume that they have similar opinions; 2) Given a set of criteria, if two participants' opinions are similar for the majority of important criteria, it is rational to presume that they have similar opinions; and 3) Given a decision problem, if two participants' opinions produce a similar decision, it is rational to presume that they have similar opinions. Because these three assumptions are presented from three similarity levels, this chapter refers the method to a three-level-similarity measuring (TLSM) method.

7.2 A gradual aggregation algorithm

7.2.1 Motivations and implementations

In this section, a gradual aggregation algorithm (GAA) is developed. The GAA is motivated by two practical issues when developing a decision support system. One concerns missing value; and the other is about generating a decision dynamically.

Evaluation aggregation used in an MCGDM problem is conventionally conducted as a one-off procedure. For effectiveness reasons, the inputs are required to be significant values in a given value set (eg. terms or numbers) and cannot have missing values. However, real evaluations cannot avoid missing values. Hence, how to process missing values is a regularly faced issue when making a decision, which has not been well solved. Another motivation for presenting the GAA is the phenomenon of generating a decision dynamically, which refers to the procedure of making a decision where a final decision is sketched based on a few numbers of criteria at the initial stage, and then amended in the following stages by considering more criteria added gradually. A typical example of generating decision dynamically is booking a flight ticket. At the beginning, a passenger has some preliminary requirements for a ticket such as the airline provider, the departure and/or arrival times. If these requirements cannot be fully met, the passenger may consider extra requirements of price, stops, etc, on top of those preliminary requirements until the find the most satisfactory ticket. In order to model this procedure quantitatively, the GAA is therefore presented.

We have currently implemented the GAA in two ways, i.e., the ordinary gradual aggregation (OGA) and the weighted gradual aggregation (WGA). The difference between them is that OGA does not explicitly consider the weights of criteria, but leaves it to the aggregation; while the WGA does. We define the OGA and WGA in Definition 7.2.1 and Definition 7.2.2 and illustrate the procedure of GAA in Figure 7.1.

Following the formal notations in Calvo *et al.* (2002a), an aggregation operator $\boldsymbol{\mathcal{A}}$ over a closed set X is denoted by $\boldsymbol{\mathcal{A}} : \bigcup_{i \in \mathbb{N}^+} \{A_i : X^i \to X\}$ where A_i is a mapping from X^i to X and is called the i-ary aggregation operator in $\boldsymbol{\mathcal{A}}$. By this notation, an aggregation operator $\boldsymbol{\mathcal{A}}$ is a family of operators with the same computational form but different number of inputs. Particularly, the unitary aggregation operator A_1 is the identity mapping. For convenience of discussion and practical demands, let X be a

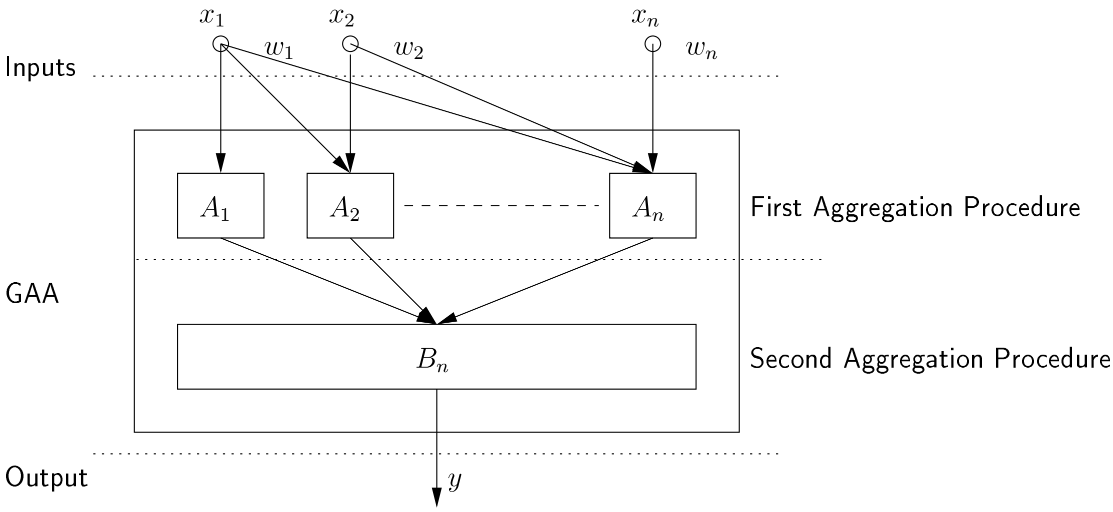


Figure 7.1: The typical GAA procedure

subset of $\mathbb{R}$.

Definition 7.2.1. *Let $\mathcal{A}$ and $\mathcal{B}$ be two aggregation operators. A mapping G_n from X^n to X is called an n-ary ordinary gradual aggregation (OGA) with respect to $\mathcal{A}$ and $\mathcal{B}$:*

$$G_n(x_1, \cdots, x_n) = B_n(\{\alpha_i | \alpha_i = A_i(x_1, \cdots, x_i), i = 1, \ldots, n\}). \tag{7.1}$$

Definition 7.2.2. *Let $\mathcal{A}$ and $\mathcal{B}$ be two aggregation operators; w_i the weight of input x_i, $i = 1, \ldots, n$. A mapping G_n from X^n to X is called an n-ary weighted gradual aggregation (WGA) with respect to $\mathcal{A}$ and $\mathcal{B}$:*

$$G_n(x_1, \cdots, x_n; w_1, \cdots, w_n) = B_n(\{\alpha_i | \alpha_i = A_i(x_1, \cdots, x_i; w_1, \cdots, w_i), i = 1, \ldots, n\}). \tag{7.2}$$

Because OGA and WGA are defined by the aggregation operators $\mathcal{A}$ and $\mathcal{B}$, they inherit some properties of $\mathcal{A}$ and $\mathcal{B}$. Some examples are given below. These properties indicate that the OGA and WGA exactly can be used to implement aggregation procedure.

Theorem 7.2.1. *If both $\mathcal{A}$ and $\mathcal{B}$ are idempotent, i.e.,*

$$A_i(x,\ldots,x)=x, \qquad\qquad B_j(x,\ldots,x)=x;$$

so do OGA and WGA.

Theorem 7.2.2. *If both $\mathcal{A}$ and $\mathcal{B}$ are monotonic, i.e.,*

$$A_i(x_1,\ldots,x_i)\leqslant A_i(y_1,\ldots,y_i) \text{ if } x_k\leqslant y_k, k=1,\ldots,i$$
$$B_j(x_1,\ldots,x_j)\leqslant B_j(y_1,\ldots,y_j) \text{ if } x_k\leqslant y_k, k=1,\ldots,j;$$

so do OGA and WGA.

Theorem 7.2.3. *If both $\mathcal{A}$ and $\mathcal{B}$ are bounded, i.e.,*

$$A_i(x_1,\ldots,x_i)\in X, \qquad\qquad B_j(x_1,\ldots,x_j)\in X;$$

So do OGA and WGA.

7.2.2 Weights assignment and adjustment

Weights of criteria are important parameters of evaluation aggregation. Assigning and/or adjusting weights in a decision problem is not an easy task (Zhang *et al.*, 2004). When developing the GAA, we noted that it can implement weights assignment and adjustment by itself when the used aggregation operators are the arithmetic mean and weighted mean. Moreover, the GAA can preserve the impacts of important criteria in the assignment and adjustment.

The OGA does not explicitly process the weights of criteria. However, when both $\mathcal{A}$ and $\mathcal{B}$ are arithmetic means, the OGA will assign weights to its inputs implicitly. Suppose a set of inputs x_1, x_2, …, x_n are indexed by their processing order, whose weights are not known. Then by the OGA, we have

$$A_1(x_1) = x_1;$$
$$A_2(x_1, x_2) = \frac{x_1 + x_2}{2};$$
$$\ldots\ldots$$
$$A_n(x_1, x_2, \ldots, x_n) = \frac{x_1 + x_2 + \ldots + x_n}{n}$$

and

$$G_n(x_1, \ldots, x_n) = \frac{\sum_{i=1}^{n} A_i(x_1, \ldots, x_i)}{n} = \sum_{i=1}^{n} x_i \left(\frac{1}{n} \sum_{j=i}^{n} \frac{1}{j} \right). \tag{7.3}$$

Let β_i be the coefficient of x_i in Eq. (7.3), i.e.,

$$\beta_i = \frac{1}{n} \sum_{j=i}^{n} \frac{1}{j}, i = 1, 2, \ldots, n. \tag{7.4}$$

The sum of β_is is

$$\beta_1 + \beta_2 + \cdots + \beta_n = 1, \tag{7.5}$$

and the order of them is

$$\beta_1 > \beta_2 > \cdots > \beta_n > 0. \tag{7.6}$$

Eq. (7.5) shows that β_1, ..., β_n form a set of weights and are assigned to the inputs implicitly. Eq. (7.6) indicates that the an input processed ahead gains a higher weight. Intuitively, this weight assignment result is consistent with a real decision procedure where the most important criteria are often processed first.

Furthermore, let us check the changes of these assigned weights with respect to the parameter n. Figure 7.2 illustrates the first five assigned weights when the number of inputs $n \leqslant 18$. It shows that each β_i is convergent with the increase of n. A conclusion is drawn from this observation that, given a large enough n, the newly added inputs will exert little affect on a sketchy decision. Since the parameter n in a real problem cannot be too large, the impacts of the most important criteria underlying the inputs—which are processed ahead—are therefore strengthened.

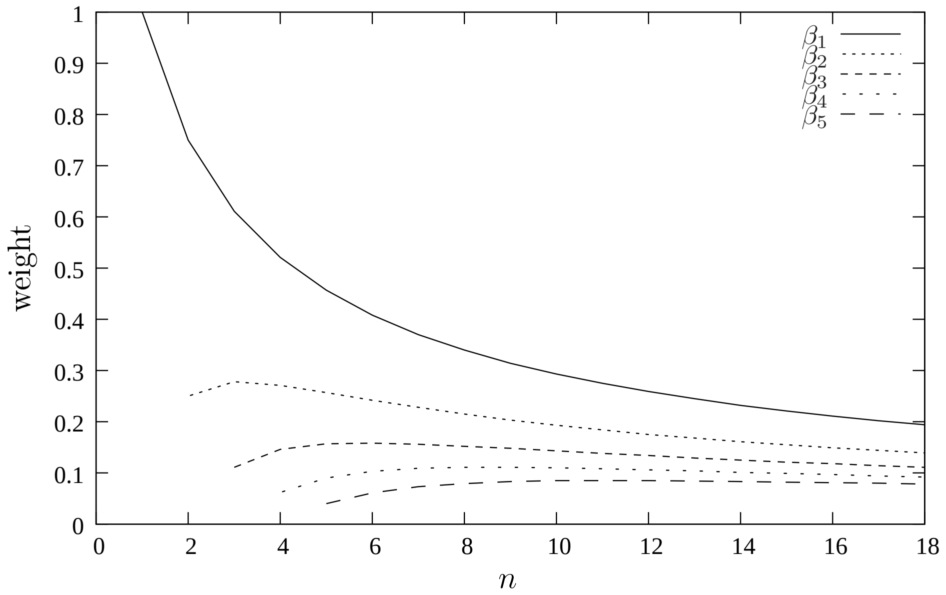


Figure 7.2: Changing weights with the number of inputs

The WGA explicitly processes the weights of criteria in its aggregation procedure. By replacing A_i with the weighted mean, and supposing the initial weight of input x_i is w_i, we noted that the WGA can adjust the initially assigned weights of the inputs. By the WGA, we have

$$A_1(x_1; w_1) = x_1$$

$$A_2(x_1, x_2; w_1, w_2) = \frac{w_1}{w_1 + w_2} x_1 + \frac{w_2}{w_1 + w_2} x_2$$

$$\cdots\cdots$$

$$A_n(x_1, \ldots, x_n; w_1, \ldots, w_n) = \frac{w_1}{\sum_{j=1}^{n} w_j} x_1 + \cdots + \frac{w_n}{\sum_{j=1}^{n} w_j} x_n$$

and

$$\begin{aligned}
&G_n(x_1, \ldots, x_n; w_1, \ldots, w_n) \\
&\quad = \frac{\sum\limits_{i=1}^{n} A_i(x_1, \ldots, x_i; w_1, \ldots, w_i)}{n} \\
&\quad = \frac{x_1 + \frac{w_1 x_1 + w_2 x_2}{w_1 + w_2} + \cdots + \sum_{i=1}^{n} \frac{w_i x_i}{\sum_{j=1}^{n} w_j}}{n} \\
&\quad = \frac{1}{n} \sum_{i=1}^{n} x_i w_i \left(\sum_{k=i}^{n} \frac{1}{\sum_{j=1}^{k} w_j} \right).
\end{aligned}$$

Let β_i be the coefficient of x_i, i.e.,

$$\beta_i = \frac{w_i}{n} \sum_{k=i}^{n} \frac{1}{\sum_{j=1}^{k} w_j}, \quad i = 1, \ldots, n \tag{7.7}$$

Then we have

$$\beta_1 + \beta_2 + \cdots + \beta_n = 1, \tag{7.8}$$

i.e., β_1, β_2, ..., β_n form a set of weights and the inputs are re-weighted by them. Comparing β_i and w_i, we have a loose inequity that

$$\beta_i \geqslant \frac{n - (i-1)}{n} w_i, \quad i = 1, \ldots, n. \tag{7.9}$$

Further analysis indicates that $\beta_1 \geqslant w_1$ and if n is large enough and i is smaller, the first several β_is are very near to, even greater than, the initial w_is. This means the impacts of those criteria are still kept by the WGA implementation.

The above algorithm and discussions indicate that the GAA can effectively maintain the impacts of important criteria. This feature is very important for our next discussion regarding making decisions dynamically and processing missing values.

7.2.3 Dynamic decision and missing values

Definitions of OGA and WGA indicate that the two implementations of GAA are closely related to the processing order of the inputs. The GAA emphasises the processing order of inputs because it is closely related to the dynamic generation of a decision and process of missing values.

In the course of making a decision, the most important criteria are often considered preferentially, then the secondary important criteria, and finally the not so important criteria. Hence, there is a natural processing order between those criteria. Similarly, there is an ordering among the inputs if they are treated as evaluations of those criteria, as shown in Section 7.2.2 where the GAA implementation assigns (reassigns) a set of decreasingly changed weights to its inputs according to their processing orders. In this sense, the GAA implementations are models of the generation of a dynamic decision.

Missing values are inevitable in real decision problems. Two intuitive strategies to handle missing values are: 1) completely ignore them; or 2) try to impute them. The GAA implementations can partially combine these. It is noted that the GAA is implemented with a parameter n. When n takes a value which is smaller than the total number of inputs, some inputs will then not be considered. Obviously, if there are missing values in the unprocessed inputs, these missing values will not affect the obtained aggregation result. However, if the missing values are not avoided, it means some evaluations about key criteria are not presented. In this situation, the GAA has an implicit functionality of assigning and/or adjusting the weights of its inputs. GAA repeatedly uses the aggregation operator $\mathcal{A}$ to calculate a set of candidate results. In this situation, the GAA assigns weights with slight differences to the inputs each time. The slightly adjustment of the weights of the inputs and the use of aggregation operator $\mathcal{B}$ to generate a final aggregation result impute the missing values implicitly. To illustrate this more clearly, consider the example below.

Example 7.2.1. *Suppose the GAA implementation is OGA with $\mathcal{A}$ and $\mathcal{B}$ being the arithmetic means and there are 10 inputs given in Table 7.1. Let the second input, i.e., 0.783 be missed. By OGA, the aggregation of the 10 inputs is 0.532. If we just ignore the missing value, the aggregation result obtained is 0.494. If we know the exact amount of the missing value, the aggregation result is 0.572. By comparing these three aggregation results, it is known that the OGA provides a more reliable result than just ignoring the missing value.*

Table 7.1: An example for processing a missing value

index	input (x_i)	value of A_i	
		without missing	with missing
1	0.840	0.840	0.840
2	<u>0.783</u>	0.812	0.42
3	0.912	0.845	0.584
4	0.335	0.718	0.522
5	0.278	0.630	0.473
6	0.477	0.604	0.474
7	0.365	0.57	0.458
8	0.952	0.618	0.519
9	0.636	0.620	0.533
10	0.142	**0.572**	**0.494**
		0.683	**0.532**

7.3 A three-level-similarity measuring method for the MOSP problem

In this section, the TLSM method for solving the MOSP problem is presented. Section 7.3.1 addresses the MOSP problem briefly. Section 7.3.2 overviews the main steps of the TLSM method. Details of those steps are introduced in Sections 7.3.3 and Section 7.3.5.

7.3.1 The MOSP problem

An MOSP problem is briefly addressed as follows. Given an MCGDM problem with some candidate options, the participants evaluate these options in terms of a set of evaluation criteria and everyone completes an evaluation report. Each participant's evaluations are summarised in linguistic terms. After collecting these evaluation reports, a question arises: can we identify which two participants have similar opinions on this kind of decision problem, based on the collected evaluation reports.

For convenience of discussion, the candidate options are denoted by $O = \{o_i | i \in I\}$; the evaluation criteria are denoted by $C = \{c_j | j \in J\}$; and the participants are denoted by $E = \{e_k | k \in K\}$. An evaluation report is denoted by a 2-D matrix $V_k = (v_{ij})_{I \times J}$, where k is the index of participant e_k and v_{ij} is the evaluation (i.e., opinion) on option o_i about criterion c_j. v_{ij} is either an element in T_j, which is the collection of all linguistic terms used by the participants for criterion c_j, or a blank for "not available" or "no answer". Without loss of generality, we suppose that each participant provides only one term for each option about each criterion.

7.3.2 Overview of the TLSM method

The outline of the TLSM method is shown in Table 7.2. By the TLSM method, the similarity of two participants' opinions will be measured at three sequential levels, i.e., the Assessment-Level, the Criterion-Level, and the Problem-Level.

At the Assessment-Level, the evaluations of two participants are compared option by option in terms of a given criterion. The comparison is conducted based on the assumption that two participants should have higher similar opinions if the number of candidate options on which they have similar evaluations is bigger. To judge whether or not two evaluations are similar, the term set T_j is firstly divided into several semantic-equal groups by pari-wise comparison on the semantics of terms used; two terms are,

Table 7.2: Outlines of main processes in the TLSM method

Process level	Main steps
Assessment-Level	*Input: two experts' evaluation reports; evaluation term set* T_j *Output: the similarity about criterion* c_j
	Step 1.1 determine a similarity matrix for evaluation terms for criterion c_j; Step 1.2 determine a clustering algorithm; Step 1.3 generate semantic-equal groups by the clustering algorithm; Step 1.4 calculate similarity between two opinions for criterion.
Criterion-Level	*Input: the similarity at the assessment level and weight* wc_j *of criterion* c_j, $j \in J$ *Output: similarity with respect to criterion* c_j *against the criteria set* c_j, $j \in J$
	Step 2.1 identify a similarity utility function u_j of criterion c_j for each $j \in J$; Step 2.2 calculate similarity with respect to criterion c_j by u_j.
Problem-Level	*Input*: similarities obtained at the criterion level *Output*: similarity between two opinions
	Step 3.1 construct the GAA from a pair of aggregation operators; Step 3.2 calculate the similarity between opinions using the GAA.

then, said to be similar (or have similar semantics) if they are in the same group. By the comparison conducted option by option on the two participants evaluations, it is known to what extent the two participants have similar opinions on a given criterion from the viewpoint of a single criterion. The similarity should be proportional to the ratio of the number of options with similar evaluations against the total number of options.

At the Criterion-Level, the differences in the weights of evaluation criteria are taken into account. The TLSM method defines for each criterion a similarity utility function based on its weight against those of other criteria. A similarity utility function meets two requirements: 1) it is proportional to similarity obtained at the Assessment-Level; and 2) it is proportional inversely to the weight for the same similarity at the Assessment-Level. The two requirements reflect the demand in practice that requirements on similarity measures of more important criteria are stricter than those of less important criteria. Based on these similarity utility functions, it is known that to what extent the two participants have similar opinions on a given criterion against a set of criteria.

At the Problem-Level, the similarity is measured using the GAA developed in Section 7.2. The GAA takes the similarities obtained at the Criterion-Level as inputs and re-orders them according to the decreasing-ordered weights of the corresponding criteria. The aggregation algorithm will generate a set of candidate values of the overall similarity of two participants opinions at the first stage, and then derives the overall similarity from them at the second stage. The overall similarity obtained indicates to what extent the two participants have similar opinions from the viewpoint of a decision problem.

The details of the TLSM method are described in the following sections.

7.3.3 Measuring similarity at the Assessment-Level

The main task at this level is to segment the term set T_j of a given criterion c_j into several semantic-equal groups. To do so, the TLSM method uses pair-wise comparison on the semantics of each pair of terms in T_j to obtain a similarity matrix; then applies a clustering algorithm, such as the Hierarchical Clustering for Fuzzy Similarity Matrix (HCFSM) (Munda, 2009), to the similarity matrix to generate semantic-equal groups.

Each element of the similarity matrix is the similarity between a pair of terms in T_j obtained by direct comparison. For a given criterion c_j, the similarity matrix S_j for the terms in T_j is denoted by

$$S_j = \begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1p_j} \\ s_{21} & s_{22} & \cdots & s_{2p_j} \\ \vdots & \vdots & \ddots & \vdots \\ s_{p_j1} & s_{p_j2} & \cdots & s_{p_jp_j} \end{pmatrix}, \qquad (7.10)$$

where

- $s_{pr} \in [0, 1]$ for any $p, r \in \{1, \dots, p_j\}$;

- $s_{rr} = 1$ for any $r \in \{1, \dots, p_j\}$;

- $s_{pr} = s_{rp}$ for any $p, r \in \{1, \dots, p_j\}$.

Remark 7.3.1. *Pair-wise comparison is used here for some practical considerations. First, linguistic terms are often represented by fuzzy sets or fuzzy numbers. The semantic interpretation of these terms varies person to person and case by case. Pair-wise comparison can avoid difficulties in the course of defining a term's semantics. Secondly, some linguistic terms are incomparable, such as colour. It is hard to define an appropriate and rational similarity measurement for these types of terms. Thirdly, similarity between some terms may be changeable. In one context, two terms may be distinguishable; however, in the other context, they are identical. Pair-wise comparison has been proved an effective strategy to analyse relationships between a set of factors; for instance, the Analytic Hierarchy Process (AHP) technique extensively uses pair-wise comparison to obtain local-priority and global-priority. Using it can better fit an application's specific setting and avoid potential heavy and complicated calculations. Of course, we do not reject other methods to determine the semantic similarity matrix.*

After obtaining the similarity matrix, the TLSM method will segment the term set by a clustering algorithm based on it. There are lots of clustering algorithms for this purpose. However, noting that the total number of terms in the term set is often between 5 and 9, i.e., it is relatively small-size, the TLSM method uses the HCFSM to implement segmenting:

- derive the transitive closure $\hat{S}_j$ from S_j by

$$\hat{S}_j = S_j \bigcup S_j^2 \bigcup S_j^4 \bigcup \cdots ,$$

where S_j^{2k} is the max-min composition of S_j^k;

- decompose $\hat{S}_j$ into a set of α-level equivalence class $(\hat{S}_j)_\alpha$ by

$$\hat{S}_j = \bigcup_{\alpha \in [0,1]} \alpha(\hat{S}_j)_\alpha;$$

- terms in T_j whose similarities belong to the same $(\hat{S}_j)_\alpha$ form a semantic-equal term group TG_j^α and are treated with similar semantic.

After segmenting T_j, two participants' opinions on criterion c_j are compared option by option. Based on the comparison result, a similarity can be defined according to the number of options on which the two opinions are treated similarly and the total number of candidate options; as a simple illustrative example, the TLSM let the similarity be the ratio of them.

Remark 7.3.2. *The segmentation of T_j is not unique. It is influenced by the parameter α, i.e., for different α, the semantic-equal groups may not be identical. The adjustable parameter α meets the real demands in applications where different parameters should be used for different criteria.*

The example below illustrates the processes in this step.

Example 7.3.1. *Suppose the similarity matrix between terms in an assessment set T is given by*

$$S = \begin{pmatrix} 1.00 & 0.89 & 0.14 & 0.29 & 0.15 & 0.34 & 0.09 \\ 0.89 & 1.00 & 0.04 & 0.24 & 0.23 & 0.55 & 0.87 \\ 0.14 & 0.04 & 1.00 & 0.09 & 0.34 & 0.20 & 0.80 \\ 0.29 & 0.24 & 0.09 & 1.00 & 0.16 & 0.08 & 0.15 \\ 0.15 & 0.23 & 0.34 & 0.16 & 1.00 & 0.31 & 0.04 \\ 0.34 & 0.55 & 0.20 & 0.08 & 0.31 & 1.00 & 0.33 \\ 0.09 & 0.87 & 0.80 & 0.15 & 0.04 & 0.33 & 1.00 \end{pmatrix},$$

By the HCFSM, the first step is to obtain the transitive closure $\hat{S}$ of S, which is

$$\hat{S} = \begin{pmatrix} 1.0 & 0.89 & 0.8 & 0.29 & 0.34 & 0.55 & 0.87 \\ 0.89 & 1.0 & 0.8 & 0.29 & 0.34 & 0.55 & 0.87 \\ 0.8 & 0.8 & 1.0 & 0.29 & 0.34 & 0.55 & 0.8 \\ 0.29 & 0.29 & 0.29 & 1.0 & 0.29 & 0.29 & 0.29 \\ 0.34 & 0.34 & 0.34 & 0.29 & 1.0 & 0.34 & 0.34 \\ 0.55 & 0.55 & 0.55 & 0.29 & 0.34 & 1.0 & 0.55 \\ 0.87 & 0.87 & 0.8 & 0.29 & 0.34 & 0.55 & 1.0 \end{pmatrix}.$$

Based on $\hat{S}$, a dendrogram is then obtained as shown in Figure 7.3. The dendrogram indicates that there are seven possible segmentation results. The 1.0 level is the strictest segmentation, by which no two terms are treated as similar, except that they are identical. The 0.87 level is looser than the 1.0 and 0.89 levels, by which the terms t_1, t_2 and t_7 can be treated as similar.

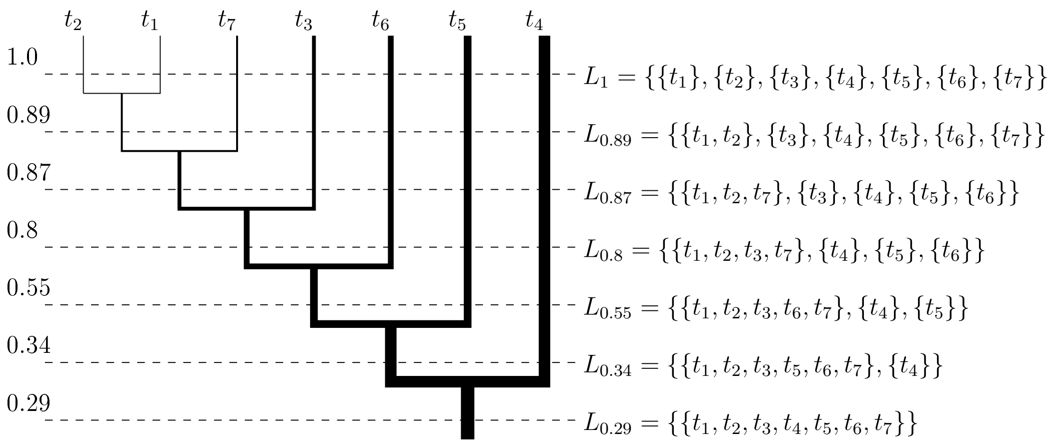


Figure 7.3: The dendrogram obtained by the HCFSM on similarity matrix S

7.3.4 Measuring similarity at the Criterion-Level

The main task in this step is identifying an appropriate similarity utility function for each criterion. To achieve this goal, two requirements are used to design a similarity utility function: 1) the function is proportional to similarity at the Assessment-Level (PSA); and 2) it is proportional inversely to the weight of a criterion (PRW).

Formally, a similarity utility function is defined below.

Definition 7.3.1. *A similarity utility function $u(nsp, w)$ of a given criterion c is a mapping from $\mathbb{N} \times W$ to $[0, 1]$ if u satisfies the PSA and PRW requirements, where $\mathbb{N}$ is the set of natural numbers and W is the range of weight.*

Functions satisfying Definition 7.3.1 are numerous. For simplicity, this study uses the following monotone and continuous function to illustrate the TLSM method:

$$u_j(nsp_j, wc_j) = \left(\frac{nsp_j}{n}\right)^{f(wc_j)} \tag{7.11}$$

where nsp_j is the number of options on which two opinions are treated as similar, n is the total number of candidate options, and $f(wc_j)$ is a parameter determined by wc_j. To finalize the similarity utility function described in Eq. (7.11), it needs to determine parameter $f(wc_j)$.

Because non-negative real numbers and linguistic terms are commonly used as weights of criteria in an MCGDM problem, we will illustrate how to finalize a similarity utility function for the two requirements.

7.3.4.1 Weights are non-negative real numbers

Suppose $wc_1, \ldots, wc_m$ is a set of normalized weights and $wc_j \geqslant 0$, $\sum_{j=1}^{m} wc_j = 1$, $m = |C|$. Without loss of generality, suppose $wc_1 \leqslant wc_2 \leqslant \cdots \leqslant wc_m$. In this situation, we determine the parameter $f(wc_j)$ as follows:

- determine a reference value wc_{j_0} and set $f(wc_{j_0}) = 1$;
- for each wc_j, set $f(wc_j) = wc_j/wc_{j_0}$.

To find a wc_{j_0} from wc_1, ..., wc_m, the following illustrative method is used: if m is odd, then set $wc_{j_0} = wc_{(m+1)/2}$; if m is even, then set $wc_{j_0} = (wc_{m/2} + wc_{m/2+1})/2$. Based on this wc_{j_0}, all wc_js are then mapped to $[0, \infty)$ by

$$f(wc_{j_0}) = 1, \quad f(wc_j) = \frac{wc_j}{wc_{j_0}}, \quad j = 1, 2, \ldots, m. \tag{7.12}$$

Remark 7.3.3. *The f used in Eq. (7.12) is just used for illustration purpose. In fact, they can be in other forms in real applications accordingly.*

To summarize the above process, let us consider a numeric example. Suppose seven criteria are considered and their weights are shown in column 2 in Table 7.3. Under this setting, the weights are listed in an increasing order as outlined below:

$$0.01 \leqslant 0.03 \leqslant 0.08 \leqslant 0.09 \leqslant 0.15 \leqslant 0.31 \leqslant 0.33$$

Because 7 (the number of criteria weights) is an odd number, the wc_{j_0} is therefore set as 0.09. Then let f be of form shown in Eq. (7.12), the parameters $f(wc_j)$s of the similarity utility function for the seven criteria can be obtained as shown in column 3 in Table 7.3.

Table 7.3: Weights and their corresponding parameters of criteria (with 2 valid positions)

criteria index j	weight wc_j	parameter $f(wc_j)$
1	0.09	1.00
2	0.01	0.08
3	0.03	0.30
4	0.31	3.44
5	0.08	0.91
6	0.33	3.68
7	0.15	1.71

7.3.4.2 Weights are linguistic terms

Linguistic weights are often represented by fuzzy numbers (or fuzzy sets). Specific numeric features of a fuzzy number (set), such as its centre of gravity (COG) or its generalized integral, can be used to determine the parameter $f(wc_j)$. A brief outline for determining this parameter is given below.

- select a numeric feature NF of fuzzy numbers and calculate NF_j of the linguistic term (i.e., a fuzzy number) wc_j;
- determine $f(NF_j)$ following steps for $f(wc_j)$ in Section 7.3.4.1;
- set $f(wc_j) = f(NF_j)$ in Eq. (7.12).

Following this outline, let us consider an illustrative example. Suppose the linguistic weights are "Very High (VH)", "Fairly High (FH)", "Medium (M)", "Rather Low (RL)", and "Very Low (VL)"; and their corresponding fuzzy numbers are shown in Figure 7.4(b) and the selected numeric feature is the horizontal coordinate of COG of a fuzzy number, i.e.,

$$NF_j = \frac{\int x\mu(x)dx}{\int \mu(x)dx} \tag{7.13}$$

where $\mu(x)$ is the membership function of the fuzzy number. By Eq. (7.13), the numeric features of these linguistic weights are calculated as shown in Table 7.4.

Following steps in Section 7.3.4.1, the $f(NF_j)$ is calculated and shown in Table 7.4. Replacing the $f(wc_j)$ in Eq. (7.12) by $f(NF_j)$, we obtain the similarity utility functions for the five linguistic weights, which can then be applied to calculate the similarity at the criterion level.

After determining the similarity utility function for each given criterion, we can apply them to measure the similarity of two participants' opinions at the Criterion-Level. For example, suppose a referential criterion is weighted "FH" and two participants' evaluations are treated similarly for seven options against a total nine options, then

Table 7.4: Numeric feature and parameter of similarity utility function of criteria (with 2 valid positions)

wc_j	VH	FH	M	RL	VL
NF	0.9	0.767	0.5	0.233	0.1
$f(NF)$	1.800	1.534	1	0.466	0.200

the similarity of these two participants' opinions with respect to this criterion is 0.680 ($= (7/9)^{1.534}$) by Eq. (7.11).

7.3.5 Measuring similarity at the Problem-Level

Section 7.3.4 details how to measure similarity of two opinions about each individual criterion from the viewpoint of a set of criteria. An individual criterion provides a single perspective by which we observe the similarity of two opinions. A set of criteria provides multiple observations. The main task in this step is to integrate those observations to form a comprehensive one. We will use the GAA developed in Section 7.2 to generate the comprehensive similarity.

The following two examples illustrate how to use the GAA. Suppose the similarities with respect to 10 criteria are obtained at the criterion level, which are 0.840, 0.783, 0.912, 0.335, 0.278, 0.477, 0.365, 0.952, 0.636, and 0.142.

Example 7.3.2. *This example illustrates the usage of OGA. Assume that both $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ are the arithmetic mean. For the 10 inputs, the GAA first generates 10 candidate similarities for the final similarity $\bar{s}$ by using A_i, where $i = 1, \ldots, 10$:*

$$\bar{s}_1 = 0.840, \quad \bar{s}_2 = 0.812, \quad \bar{s}_3 = 0.845, \quad \bar{s}_4 = 0.718,$$
$$\bar{s}_5 = 0.630, \quad \bar{s}_6 = 0.604, \quad \bar{s}_7 = 0.570, \quad \bar{s}_8 = 0.617,$$
$$\bar{s}_9 = 0.619, \quad \bar{s}_{10} = 0.572.$$

Then the GAA applies B_{10} to the 10 candidate similarities $\bar{s}_1$, …, $\bar{s}_{10}$ to generate $\bar{s}$

which is $\bar{s} = 0.683$*, i.e., the similarity of the two experts' opinions is* 0.683*.*

Example 7.3.3. *This example illustrates the usage of WGA. Assume that* $\mathcal{A}$ *is the OWA aggregation and* $\mathcal{B}$ *is the arithmetic mean. Because an OWA aggregation needs the weights of inputs, we randomly generate 10 weights for them as:*

$$w_1 = 0.394, \quad w_2 = 0.798, \quad w_3 = 0.198, \quad w_4 = 0.768,$$
$$w_5 = 0.554, \quad w_6 = 0.629, \quad w_7 = 0.513, \quad w_8 = 0.916,$$
$$w_9 = 0.717, \quad w_{10} = 0.607.$$

Then, GAA calculates the candidate values of $\bar{s}_i$*s following OWA:*

$$\bar{s}_1 = 0.952, \quad \bar{s}_2 = 0.925, \quad \bar{s}_3 = 0.913, \quad \bar{s}_4 = 0.866,$$
$$\bar{s}_5 = 0.819, \quad \bar{s}_6 = 0.755, \quad \bar{s}_7 = 0.703, \quad \bar{s}_8 = 0.632,$$
$$\bar{s}_9 = 0.586, \quad \bar{s}_{10} = 0.541.$$

Finally, GAA applies the B_{10} *to* $\bar{s}_1$*, ...,* $\bar{s}_{10}$ *to find the overall similarity, which is* 0.769*.*

Based on the similarity measurement at the three levels, an overall similarity between two participants' opinions is generated, which can be used as the answer of the MOSP problem. In the next section, we will apply the TLSM method to two real-world problems.

7.4 Case studies

This section applies the TLSM method to an social policy selection application and an energy policy evaluation application.

7.4.1 Do similarities exist between social actors?

This example is quoted from Munda (2008, 2009). In a social policy selection problem, six social actors (i.e., participants) have presented their assessments for seven possible policies (i.e., options). The social impact matrix (i.e., evaluation report) is given in Table 7.5 and the semantics of the used linguistic assessments are given in Figure 7.4(a). The problem is to answer whether or not similarities exist between these social actors.

Table 7.5: An illustrative example of social impact matrix

Social	Policy options						
actors	a_1	a_2	a_3	a_4	a_5	a_6	a_7
b_1	Very good	Good	Moderate	bad	Fairly good	Fairly bad	Very bad
b_2	Very good	Good	Moderate	Bad	Fairly good	Very bad	Very bad
b_3	Very bad	Fairly bad	Moderate	Good	Very good	Good	Moderate
b_4	Very bad	Fairly bad	Fairly bad	Good	Fairly good	Good	Very good
b_5	Very bad	Bad	Fairly bad	Moderate	Fairly good	Good	Very good
b_6	Very bad	Good	Bad	Good	Good	Good	Very good

Firstly, we recited the solution in Munda (2009) as a comparison with the TLSM method. The Munda's method includes three main steps.

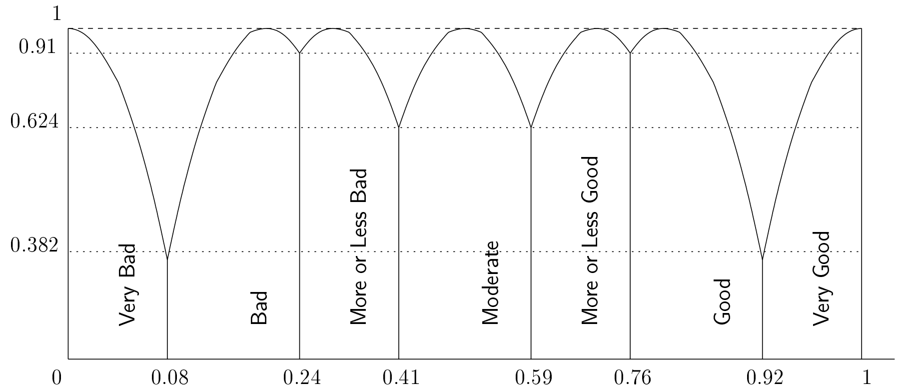
- Generate a similarity matrix between the social actors by a similarity measurement of linguistic assessments:

$$s(b_i, b_j) = \frac{1}{1 + \left[\sum_{k=1}^{7} \left(\iint_{x,y} |x - y| f_i(x) g_j(y) \mathrm{d}y \mathrm{d}x \right)^2 \right]^{1/2}}.$$

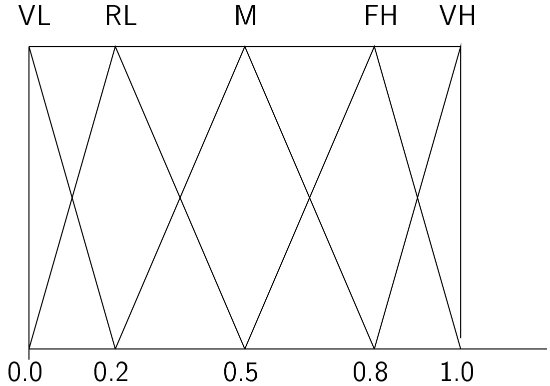
  where $\mu_1(x)$ and $\mu_2(y)$ are membership functions of two linguistic terms (as assessments) x and y, respectively; and $d(x, y) = \iint_{x,y} |x - y| f(x) g(y) \mathrm{d}y \mathrm{d}x$ is the semantic distance between x and y.

  By this measurement, the similarity matrix S of the six social actors is obtained, and is presented in Table 7.6.

- Generate hierarchical clustering. By using the HCFSM clustering algorithm, a dendrogram is given in Figure 7.5(a).



(a) Linguistic assessments in Munda's method.



(b) Linguistic weights in Case 2.

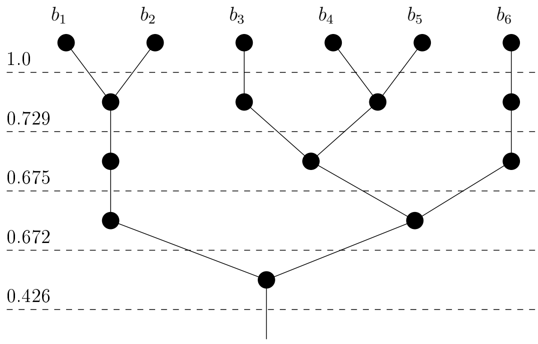
Figure 7.4: Semantic of linguistic terms

Table 7.6: Similarity matrix between six social actors

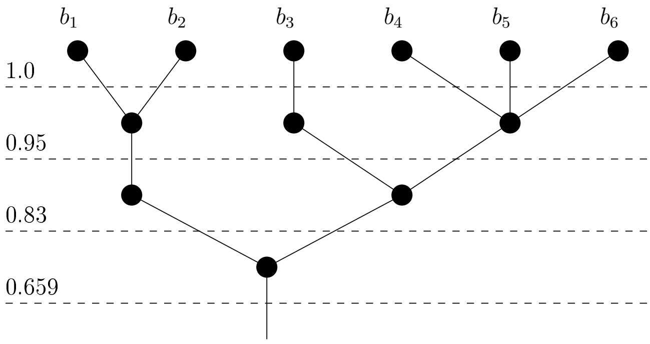
	b_1	b_2	b_3	b_4	b_5	b_6
b_1	1	0.729	0.426	0.399	0.403	0.403
b_2	0.729	1	0.410	0.386	0.390	0.390
b_3	0.426	0.410	1	0.675	0.584	0.569
b_4	0.399	0.386	0.675	1	0.729	0.672
b_5	0.403	0.390	0.584	0.729	1	0.595
b_6	0.403	0.390	0.569	0.672	0.595	1

- Analyze clustering result. By the clustering result, the social actors b_1 and b_2 have higher similarity.

We now apply the presented TLSM method to resolve this problem. For convenience, we take the social actors b_1 and b_4 as examples to illustrate the experiment. Moreover, because the problem setting does not provide any information about evalu-



(a) Result by Munda's method.



(b) Result by the TLSM method.

Figure 7.5: Dendrogram of similarities between experts

ation criteria, we can assume that it only concerns one criterion.

Step 1: Measuring similarity at the Assessment-Level.

Firstly, we use the following distance measure between two terms t_i and t_j to obtain the similarity matrix of all linguistic assessments:

$$d(t_i, t_j) = |x_i - x_j|, \tag{7.14}$$

where x_i and x_j are the points whose membership degrees are equal to 1 with respect to t_i and t_j respectively. Based on this distance, the similarity between t_i and t_j is defined by

$$s_{ij} = 1 - d(t_i, t_j). \tag{7.15}$$

Therefore the similarity matrix for linguistic assessments is obtained and shown in Table 7.7. Hence, the dendrogram for the seven linguistic assessments by the HCFSM clustering algorithm is obtained and presented in Figure 7.6.

Table 7.7: Similarity matrix for linguistic assessments

Term	Very bad	Bad	Fairly bad	Moderate	Fairly good	good	Very good
Very bad	1.0	0.8	0.7	0.5	0.3	0.2	0.0
Bad	0.8	1.0	0.9	0.7	0.5	0.4	0.2
Fairly bad	0.7	0.9	1.0	0.8	0.6	0.5	0.3
Moderate	0.5	0.7	0.8	1.0	0.8	0.7	0.5
Fairly good	0.3	0.5	0.6	0.8	1.0	0.9	0.7
good	0.2	0.4	0.5	0.7	0.9	1.0	0.8
Very good	0.0	0.2	0.3	0.5	0.7	0.8	1.0

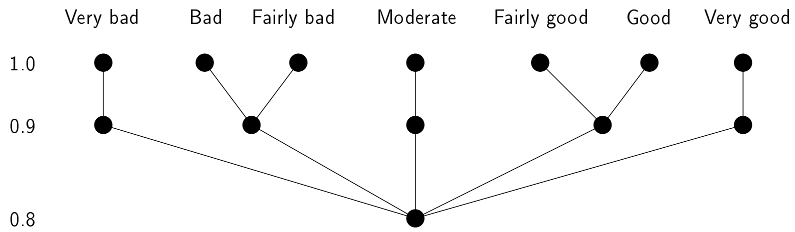


Figure 7.6: Dengrogram of linguistic assessments (terms)

We next take 0.9-level equivalence-class in Figure 7.6 to segment the seven terms and compare the evaluations from actors b_1 and b_4. It is noted that these two social actors have a similar opinion on policy a_5 only. Table 7.8 lists the number of options on which participants have similar opinions by pair-wise comparison.

Table 7.8: Number of options with similar opinions by pairwise comparison

nsp	b_1	b_2	b_3	b_4	b_5	b_6
b_1	7	6	1	1	1	2
b_2	6	7	1	1	1	2
b_3	1	1	7	4	3	3
b_4	1	1	4	7	6	6
b_5	1	1	3	6	7	5
b_6	2	2	3	6	5	7

Step 2: Measuring similarity at the Criterion-Level. Because this problem involves only one criterion, it is enough to determine a unique parameter $f(wc)$.

For simplicity, suppose the similarity utility function used is of the form in Eq.

(7.11). Setting $f(wc)$ to be less than, equal to, or greater than 1.0 obtains three typical utilities of a criterion. The three utilities are illustrated below respectively.

The first situation is setting $f(wc) = 1$. The similarity utility function is, therefore, a linear function, by which the similarity between b_1 and b_4 is 0.143. Table 7.9 illustrates the pair-wise similarity of all actors under this setting.

Table 7.9: Pair-wise comparison of similarity at the Criterion-Level ($f(wc) = 1$)

$f(n(v))$	b_1	b_2	b_3	b_4	b_5	b_6
b_1	1	0.857	0.143	0.143	0.143	0.286
b_2	0.857	1	0.143	0.143	0.143	0.286
b_3	0.143	0.143	1	0.571	0.429	0.429
b_4	0.143	0.143	0.571	1	0.857	0.857
b_5	0.143	0.143	0.429	0.857	1	0.714
b_6	0.286	0.286	0.429	0.857	0.714	1

The second situation is setting $f(wc) > 1$. The obtained similarity utility function increases slowly with a smaller similarity at the Assessment-Level and then increases quickly with a larger one. Suppose $f(wc) = 2$, then the pair-wise similarities of the six actors are shown in Table 7.10.

Table 7.10: Pairwise comparison of similarity at the Criterion-Level ($f(wc) = 2$)

$f(n(v))$	b_1	b_2	b_3	b_4	b_5	b_6
b_1	1	0.735	0.020	0.020	0.020	0.082
b_2	0.735	1	0.020	0.020	0.020	0.082
b_3	0.020	0.020	1	0.327	0.184	0.184
b_4	0.020	0.020	0.327	1	0.735	0.735
b_5	0.020	0.020	0.184	0.735	1	0.510
b_6	0.082	0.082	0.184	0.735	0.510	1

The third situation is $f(wc) < 1$. Under this setting, the obtained similarity utility function increases quickly with a smaller similarity at the Assessment-Level and then increases slowly with a bigger one. When setting $f(wc) = 1/3$, the pair-wise similarities are shown in Table 7.11.

Based on the identified similarity utility function, the similarity between b_1 and b_4

Table 7.11: Pairwise comparison of similarity at the Criterion-Level ($\alpha = 1/3$)

$f(wc)$	b_1	b_2	b_3	b_4	b_5	b_6
b_1	1	0.950	0.523	0.523	0.523	0.659
b_2	0.950	1	0.523	0.523	0.523	0.659
b_3	0.523	0.523	1	0.830	0.754	0.754
b_4	0.523	0.523	0.830	1	0.950	0.950
b_5	0.523	0.523	0.754	0.950	1	0.894
b_6	0.659	0.659	0.754	0.950	0.894	1

is obtained at the Criterion-Level.

Step 3: Measuring similarity at the Problem-Level. Because the example only involves a unique criterion, this step is redundant, i.e., the similarity at the Criterion-Level can be used at the subject level. Therefore, the similarity between b_1 and b_4 has already been obtained, i.e. 0.020.

Noting that Table 7.11 is a similarity matrix of the six social actors, we can use the HCFSM to obtain a similar dendrogram (Figure 7.5(b)). Comparing these two dendrograms, we recognized two minor differences: 1) social actor b_6 will join the group of b_4 and b_5 earlier than social actor b_3; and 2) the parameter α is slightly different.

7.4.2 Energy policy selection with missing assessments

A governmental consultant committee has designed some energy policies for a nation's sustainable development in the future. Three of them are sent to six domain experts for evaluation in terms of eight primary criteria. Each primary criterion is composed of a few secondary criteria and the total number of criteria really evaluated is 16. An expert's evaluation report includes two components: 1) the assessments on the importance of all primary criteria concerned for sustainable development; and 2) the assessments on the impacts of the three alternative policies on sustainable development according to those criteria. All assessments are selected from a set of provided linguistic terms, or left blank for "unavailable", or with a question mark for "uncertain assessments (unknown or unsure)". After collecting the evaluation reports from these experts, the committee wants to know which two experts have similar opinions.

Without loss of generality, this study assumes that the weights of those 16 evaluation criteria have been determined in advance and the only task is to measure the similarity of the six experts. To illustrate our process more clearly, let O_1,O_2,O_3 be the three alternative policies; $c_1, \cdots ,c_{16}$ be the 16 evaluation criteria; and $e_1, \ldots, e_6$ be the six experts. The collected evaluation reports (only the assessments section in a real report) are listed in Table 7.12. The linguistic terms used in Table 7.12 for weights of criteria and evaluations on policies are summarised in Table 7.13.

Noting that all weights and assessments of the six experts are expressed by linguistic terms, this study uses triangular normal fuzzy numbers to represent linguistic terms. The semantic definitions of those linguistic terms are shown in the fourth column in Table 7.13 and in Figure 7.4(b). Based on this pre-process, the TLSM method is applied to this case and detailed steps are illustrated below.

Step 1: Measuring similarity at the Assessment-Level. To determine a similarity matrix for assessment terms, this study uses the same method shown in case 1 to define similarity between linguistic terms. The obtained similarity matrix S is

s_{ij}	AC	VL	L	UL	HUL
AC	1.0	0.8	0.5	0.2	0.0
VL	0.8	1.0	0.7	0.4	0.2
L	0.5	0.7	1.0	0.7	0.5
UL	0.2	0.4	0.7	1.0	0.8
HUL	0.0	0.2	0.5	0.8	1.0

By applying the HCSFM algorithm to S, we obtain three possible segments :

Table 7.12: Evaluation reports of six experts

c_i	w_i	O_1	O_2	O_3	O_1	O_2	O_3	O_1	O_2	O_3
		Expert 1			Expert 2			Expert 3		
1	VH	UL	L	AC	VL	VL	L	HUL	L	VL
2	FH	L	L	AC	UL	L	L	UL	UL	L
3	FH	UL	L	VL	UL	HUL	L	HUL	L	VL
4	FH	HUL	VL	AC	UL	UL	L	HUL	UL	HUL
5	FH	L	L	VL	L	VL	L	UL	VL	VL
6	FH	AC	VL	AC	VL	VL	UL	L	VL	AC
7	FH	L	UL	VL	UL	HUL	L	HUL	L	
8	FH	VL	L	VL	AC	AC	AC	UL	VL	VL
9	FH	AC	VL	L	AC	AC	AC	UL	VL	AC
10	FH	L	UL	L	VL	L	L	VL	VL	UL
11	FH	UL	UL	?	L	L	VL	VL	VL	HUL
12	FH	HUL	UL	L	HUL	HUL	VL	AC	AC	L
13	VH							UL	VL	UL
14	VH	VL	VL	VL	VL	VL	VL		VL	UL
15	FH	UL	HUL	VL	HUL	HUL	UL	L	HUL	HUL
16	FH	UL	UL	L	HUL	HUL	L	L	VL	L
c_i	w_i	O_1	O_2	O_3	O_1	O_2	O_3	O_1	O_2	O_3
	w_i	Expert 4			Expert 5			Expert 6		
1	VH	VL	L	UL	VL	UL	HUL	L	UL	HUL
2	FH	VL	L	VL	VL	UL	HUL	VL	L	UL
3	FH	AC	UL		VL	UL	HUL	L	HUL	HUL
4	FH	L	L	HUL	L	HUL	HUL	UL	UL	HUL
5	FH	AC	L	UL	AC	L	HUL	VL	VL	L
6	FH	UL	UL	HUL	AC	UL	HUL	L	UL	HUL
7	FH	UL	HUL	HUL	UL	L	HUL	HUL	HUL	HUL
8	FH	AC	VL	L	AC	UL	HUL	AC	AC	AC
9	FH	VL	AC	L	AC	UL	HUL	AC	AC	AC
10	FH	VL	L	AC	VL	UL	HUL	HUL	HUL	HUL
11	FH	HUL	HUL	L	L	UL	HUL			
12	FH	AC	AC	VL	VL	UL	HUL	L	UL	HUL
13	VH	VL	L	UL	AC	L	UL	L	HUL	HUL
14	VH	VL	VL	UL	VL	L	UL			
15	FH			UL	VL	UL	HUL	VL	UL	HUL
16	FH	UL	UL	HUL	VL	L	UL	L	UL	HUL

Table 7.13: Symbols and Semantics of linguistic terms in evaluation reports

Abbreviation.	Names	Semantics
VH	Very high	$(0.7, 1.0, 1.0)$
FH	Fairly high	$(0.5, 0.8, 1.0)$
M	Medium	$(0.2, 0.5, 0.8)$
RL	Rather low	$(0.0, 0.2, 0.5)$
VL	Very low	$(0.0, 0.0, 0.3)$
AC	Almost certain	$(0.7, 1.0, 1.0)$
VL	Very likely	$(0.5, 0.8, 1.0)$
L	Likely	$(0.2, 0.5, 0.8)$
UL	Unlikely	$(0.0, 0.2, 0.5)$
HUL	Highly Unlikely	$(0.0, 0.0, 0.3)$
NA	No answer	

segment level	Segments
1.0	{AC}, {VL}, {L}, {UL}, {HUL}
0.8	{AC, VL}, {L}, {UL, HUL}
0.7	{AC, VL, L, UL, HUL}

It is noted that the only two possible weights are used for the 16 criteria, i.e., "VH" and "FH", the segments with 1.0-level is used for criteria with weight "VH"; and the segments with 0.8-level is used for criteria with weight "FH". (Note. The segments with 0.7-level will not be used in this study because it lacks capability to distinguish different terms.) Therefore, we can compare experts' opinions at the assessments level. The following illustration will take experts e_1 and e_2 as an example.

For criterion c_1: Because the weight of c_1 is "VH", two assessments are similar if and only if they are the same one. Hence, the number of assessments with similar semantics between (UL, L, AC) (of e_1) and (VL, VL, L) (of e_2) about this criterion is 0.

For criterion c_2: Because the weight of c_2 is "FH", the assessment "AC" is treated the same as "VL"; so do "UL" and "HUL". Hence, the number of assessments with similar semantics between (L, L, AC) (of e_1) and (UL, L, L) (of e_2)) about this criterion is 1 because the two opinions have the same assessment on policy O_2 only.

Similarly, we can compare these two experts on the remaining 14 criteria one by one. Table 7.14 lists the number of options with similar opinion for all 16 criteria.

Table 7.14: Number of similar options for 16 criteria of e_1 and e_2

c_i	1	2	3	4	5	6	7	8
no. of similar assessments	0	1	1	1	1	2	1	2
c_i	9	10	11	12	13	14	15	16
no. of similar assessments	2	1	0	2	0	3	2	3

Remark 7.4.1. *It is noted that criteria c_9 and c_{13} are different from other criteria because of the missing or uncertain assessments. To deal with these missing assessments, this study treats them as dissimilar.*

Step 2: Measuring similarity at the Criterion-Level. For simplicity, this study uses the similarity utility function defined in Eq. (7.11). The parameter $f(wc_j)$ is determined by the same method as used in case 1. The numeric feature of these five linguistic terms are:

$$\begin{aligned} NF_{VH} &= 0.9, \quad NF_{FH} = 0.767, \quad NF_M = 0.5 \\ NF_{RL} &= 0.233, \quad NF_{VL} = 0.1. \end{aligned} \tag{7.16}$$

The study sets $f(NF_M) = 1.0$ and calculates the parameters for the other four weights accordingly:

$$\begin{aligned} f(NF_{VH}) &= 0.9/0.5 = 1.8, \quad f(NF_{FH}) = 1.534, \\ f(NF_{RL}) &= 0.466, \qquad f(NF_{VL}) = 0.2. \end{aligned} \tag{7.17}$$

Once similarity utility functions of all evaluation criteria are finalized, they can be used to obtain similarity at the Criterion-Level. For instance, consider the criteria c_1 and c_6. The weight of c_1 is "VH" and $f(NF_{VH}) = 1.8$; hence the similarity with respect to c_1 is 0.134. Because the weight of c_6 is "FH" and the $f(NF_{FH}) = 1.534$, then the similarity with respect to c_6 is 0.537. For the other 14 criteria, the calculation is similar. The similarities at the Criterion-Level between e_1 and e_2 are summarized below.

$$s_1 = 0.000, \quad s_2 = 0.185, \quad s_3 = 0.185, \quad s_4 = 0.185,$$
$$s_5 = 0.185, \quad s_6 = 0.537, \quad s_7 = 0.185, \quad s_8 = 0.537,$$
$$s_9 = 0.537, \quad s_{10} = 0.185, \quad s_{11} = 0.000, \quad s_{12} = 0.537,$$
$$s_{13} = 0.000, \quad s_{14} = 1, \quad s_{15} = 0.537, \quad s_{16} = 1.$$

Step 3: Measuring similarity at the Problem-Level. The GAA is implemented as follows:

- re-order the criteria by their weights in descending order;
- set A_i to be the arithmetic mean, $i = 1, \ldots, 16$;
- set B_{16} to be the t-conorm maximum $\max$.

To re-order the criteria, this study used the NF values obtained at the Criterion-Level as the ordering reference. Then following the order of criteria, the i-ary aggregation operator A_i is applied to those similarities to obtain possible similarities between the two experts:

$$0.000, 0.000, 0.333, 0.296, 0.274, 0.259, 0.249, 0.285,$$
$$0.274, 0.300, 0.322, 0.310, 0.304, 0.320, 0.362, 0.362$$

From them the biggest is selected by B_{16}, which is 0.362. Therefore, the similarity between the experts e_1 and e_2 is 0.362.

Table 7.15 gives the pair-wise similarity of the six experts. Based on the pair-wise similarity measurement, the experts can be grouped again based on a clustering method. For instance, Figure 7.7 is the dendrogram that uses the HCFSM algorithm.

Further observation indicates that experts e_4, e_5, and e_6 have higher similarities in their opinions.

Table 7.15: Pair-wise similarities of all six experts

	e_1	e_2	e_3	e_4	e_5	e_6
e_1	1	0.362	0.273	0.289	0.108	0.151
e_2	0.362	1	0.275	0.277	0.189	0.379
e_3	0.273	0.275	1	0.253	0.199	0.239
e_4	0.289	0.277	0.253	1	0.493	0.337
e_5	0.108	0.189	0.199	0.493	1	0.482
e_6	0.151	0.379	0.239	0.337	0.482	1

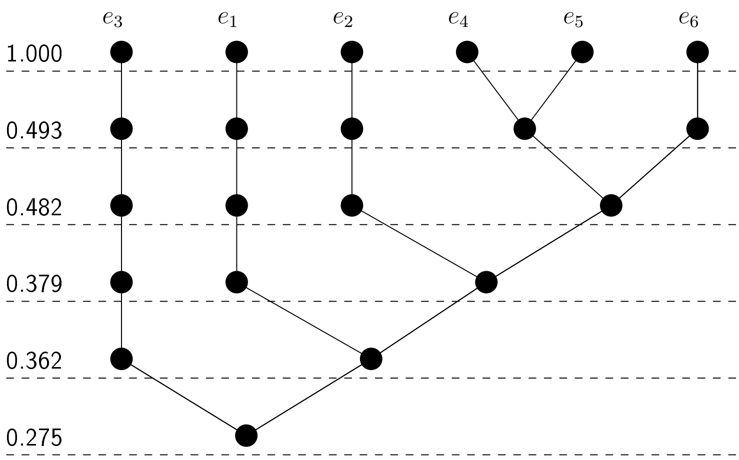


Figure 7.7: Deprogram of experts using the HCFSM

Chapter 8 Prototype of a Decision Information Processing System

This chapter illustrates the processing model, the structure, and the functionalities of a multi-criteria decision support system prototype.

Multi-criteria group decision-making (MCGDM) is an important type of group decision-making. It aims to support preference-based decision over the available alternatives that are characterized by multiple criteria in a group. To increase the level of overall satisfaction for the final decision across the group and deal with uncertainty in decision process, a fuzzy MCGDM processing (FMP) model is established in this study. This FMP model aggregates both subjective and objective information under multi-level hierarchies of criteria and evaluators. Based on the FMP model, a fuzzy MCGDM decision support system (called Decider) is developed, which can handle information expressed in linguistic terms, Boolean values, and numeric values to assess and rank a set of alternatives within a group of decision makers. Real applications indicate that the presented FMP model and the Decider software are able to effectively handle uncertainties in both subjective and objective information and support group decision-making under multi-level criteria with a higher level of satisfaction by decision makers.

The remaining sections of the chapter are organized as follows. Section 8.1 briefly introduces the background of developing such a decision support system. In Section 8.2, a fuzzy MCGDM process (FMP) model, which includes a fuzzification method and a fuzzy aggregation method, is presented and discussed. Section 8.3 describes a fuzzy multi-criteria group decision support system (DECIDER) which implements the FMP model. An application example of DECIDER is also illustrated.

8.1 Introduction

Multi-criteria decision-making (MCDM) refers to making preference decision (e.g., evaluation, prioritization, and selection) over the available alternatives that are characterized by multiple, usually conflicting, criteria. As decision-making requires multiple perspectives of different people, most organizational decisions are made in groups. Group decision-making (GDM) is the process of arriving at a judgment or a solution for a decision problem based on the input and feedback of multiple individuals. In general, a group satisfactory solution (final decision) is one that is most acceptable by the group of individuals as a whole. Since selecting a satisfactory solution affects organizational performance, it is crucial to make the group decision-making process as efficient and effective as possible. MCGDM combines MCDM and GDM methods and has been proved to be a very effective technique to increase the overall satisfactory level for the final decision across the group and particularly in evaluation decision-making such as evaluating products, developing policies, selecting employees, and arranging various resources.

MCGDM needs to consider two important hierarchies, i.e., the hierarchy of criteria and the hierarchy of evaluators (group members). A complicated decision problem is often divided into small decision problems and then on until detailed criteria are identified. To make a decision, decision information is integrated from bottom (detailed criteria) to top (the decision problem) step by step along the hierarchy of criteria. Decision information is mostly provided by evaluators who are also organized in a hierarchy. The group decision is the most acceptable alternative by all evaluators in that group as a whole.

In practice, subjective and objective information needs to be processed simultaneously in an MCGDM problem. Subjective information is mainly collected from evaluators and is often expressed by natural language, such as linguistic terms. Objective information is mainly collected from various instruments and often indicates a certain facts, such as readings from sensors or machines.

Both subjective information and objective information involves uncertainties. Linguistic terms often represent a certain degrees of uncertain judgments or concepts. For example, linguistic terms "important" and "unimportant" are uncertain concepts because the boundary between them is unclear. Another example of uncertain concepts is description "a tall man". As for objective information, uncertainty is still not easy to find out. For instance, the reading "$35°C$" of a temperature sensor is of uncertainty. Does it mean that the temperature is exactly $35°C$ and cannot be $34.9°C$ or $35.1°C$? Hence, uncertainty in both subjective and objective information needs to be processed when solving an MCGDM problem, in particular, when the decision is made in a situation with multiple information sources.

Fuzziness is an important type of uncertainty in subjective and objective information. Processing methods based on fuzzy sets and fuzzy logic have been recognized as effective tools to deal with uncertainty with fuzzy feature [1]. To effectively process fuzziness in subjective and objective information, a fuzzification and an aggregation are two widely-used techniques. A fuzzification method transfers a numeric value to a fuzzy set; while an aggregation method projects a set of inputs to an output. Because subjective information with fuzziness is often expressed by fuzzy sets, processing uncertainty in subjective information can be conducted by linguistic methods. Fuzziness in objective information can also be processed by fuzzy-set-based processing methods once objective information is transferred to fuzzy sets. In this study, a fuzzification method for transferring objective information into fuzzy numbers (a special fuzzy set)

[1]Fuzziness, possibility, reliability as well as probability are uncertainties with different natures.

is discussed in Section 8.2. Moreover, a fuzzy aggregation method is developed to generate decision reference step by step from bottom to top through the hierarchies of criteria and evaluators.

In this study, a fuzzy multi-criteria group decision support system, called DECIDER, is designed and developed. DECIDER can deal with subjective and objective information, expressed in linguistic terms or numeric values, at the same time. It is developed handling MCGDM problems with multi-level hierarchies of criteria and evaluators and uncertain information; and implementing cross-platform applications.

8.2 A fuzzy MCGDM processing model

This section introduces a fuzzy MCGDM processing (FMP) model and its advanced features. Figure 8.1 illustrates the main components of the model and the information flows in it.

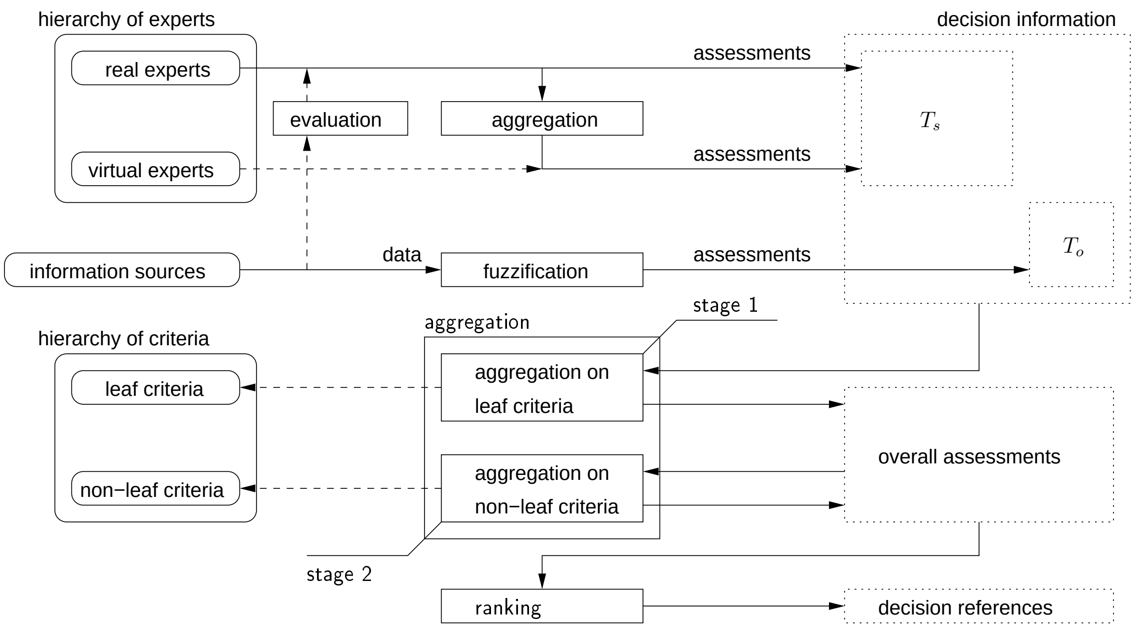


Figure 8.1: The FMP model for fuzzy MCGDM problem

8.2.1 A typical processing model for MCGDM

A typical MCGDM problem is composed of a set of alternatives (e.g., actions and policy choices), a one-level hierarchy of decision criteria, a set of decision matrices (i.e., decision tables for alternatives), as well as a set of evaluators who present those decision matrices. Three main steps are involved in selection of the best alternative(s) (Triantaphyllou, 2000):

(1) Determine the relevant criteria and alternatives;

(2) Evaluate the relative impacts of alternatives on those criteria;

(3) Determine a ranking of each alternative.

Roughly, an MCGDM processing model can be expressed as:

$$\mathcal{M} = (\mathcal{C}, \mathcal{E}, \mathcal{A}, \mathcal{T}), \tag{8.1}$$

where

- $\mathcal{C} = \{(c_j, wc_j) | j = 1, 2, \ldots, n\}$ is a set of criteria and their weights (importance, impacts);
- $\mathcal{E} = \{(e_k, we_k) | k = 1, 2, \ldots, m\}$ is a set of evaluators and their weights (impact on final decision);
- $\mathcal{A} = \{a_i | i = 1, 2, \ldots, p\}$ is a set of alternatives;
- $\mathcal{T} = \{T_i = (v^i_{jk})_{n \times m} | i = 1, 2, \ldots, p\}$ is a set of decision matrices. T_i is the decision matrix for alternative a_i, v^i_{jk} is the assessment on criterion c_j by evaluation e_k.

For each a_i, the overall assessment y_i based on $\mathcal{M}$ is explicitly expressed as

$$y_i = (wc_1, \cdots, wc_n) \circ \begin{pmatrix} v_{11}^i & v_{12}^i & \cdots & v_{1m}^i \\ v_{21}^i & v_{22}^i & \cdots & v_{2m}^i \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1}^i & v_{n2}^i & \cdots & v_{nm}^i \end{pmatrix} \diamond \begin{pmatrix} we_1 \\ we_2 \\ \vdots \\ we_m \end{pmatrix}$$

where $\circ$ and $\diamond$ are aggregation operators in $\mathcal{M}$. In such a model, a decision is made through evaluation and aggregation for collected information. Generally, model $\mathcal{M}$ has the following features:

(1) It only takes one level of criteria into account rather than a multi-level hierarchy of criteria.

(2) It only handles one level of evaluators rather than an organization hierarchy of evaluators.

(3) It does not consider the impact of information sources.

(4) It only deals with certain data.

(5) It mainly deals with subjective data.

Focuses on these features, the FMP model is presented in next section.

8.2.2 Decision context and decision information

The FMP model is presented on the following considerations: 1) it can handle uncertain information, which exists in both subjective terms and objective values; 2) a decision can be made by a multi-level hierarchy of evaluators; 3) alternatives are evaluated along a multi-level hierarchy of criteria; and 4) the nature of identified information sources impacts on decision-making. Based on these considerations, a decision context in the FMP model includes two multi-level hierarchies of evaluators and criteria, respectively; identified information sources with various natures; as well as alternatives of the decision problem.

Decision information includes subjective and objective information. In the FMP model, decision information is used to generate final decision, which comes from two channels, i.e., the evaluators and identified instruments. In general, subjective information is presented by evaluators and objective information is provided by identified instruments. In the following, the FMP model uses the term of assessment for subjective information and the term of data for objective information. The FMP model uses fuzzy numbers to represent assessments from evaluators or fuzzification results of data and uses the term "decision information" specifically refers to information which is expressed by fuzzy number. Different from decision information, the FMP model refers supporting information to those which describes the attributes/status of alternatives. For example, the sensor reading "35°C" is supporting information (also objective information). While fuzzy set $\widetilde{35}$ deduced from "35°C" is decision information.

The FMP model mainly focuses on the fuzziness in decision information process. Both subjective and objective information is expressed by fuzzy numbers on real interval [0, 1], i.e.,

$$I_s \in \mathscr{F}_{[0,1]}, \quad I_o \in \mathscr{F}_{[0,1]}, \tag{8.2}$$

where $\mathscr{F}_{[0,1]}$ is the set of fuzzy numbers on interval [0, 1], I_s and I_o represent a piece of subjective information and objective information respectively.

8.2.3 Multi-level hierarchies for evaluators and criteria

A group of evaluators are involved in complex decision context and play different roles. A multi-level hierarchy of evaluators is used to depict organization hierarchy of evaluators. The FMP model uses a tree E for the hierarchy of evaluators. The leaf nodes of tree E are evaluators, called real evaluators (REs), who are persons really present assessments on alternatives. All non-leaf nodes of tree E are group evaluators (GEs) which represent the groups of a set of REs or GEs. A typical example of hierarchy of evaluators is the staff hierarchy of a faculty in a university. For convenience, the following sections will use e_{ij} for the j-th evaluator under GE e_i.

Similar to hierarchy of evaluators, multi-level hierarchy of criteria exists because one-level hierarchy of criteria in model $\mathcal{M}$ can be extended to multi-level by adding a virtual root node which represents the decision target. A multi-level hierarchy of criteria is also expressed by a tree C. In this tree, the leaf nodes are criteria for which decision information is provided; and the non-leaf nodes are criteria acting as summary of relevant criteria. In tree C, c_{ij} is used for the j-th sub-criterion of criterion c_i.

8.2.4 Information sources and connecting strength

In the FMP model, a listing of information sources is identified, which are connected to a set of leaf criteria. An information source provides the same supporting information to the connected criteria. For different criteria, the same supporting information may produce different influences. A connecting strength is defined to measure the degree of influence. Connections R between criteria hierarchy C and list of information sources S are formally defined as:

$$R \subseteq L(C) \times S, \tag{8.3}$$

where $L(C)$ is the set of leaf nodes in C, and each $r = (c, s) \in R$ is called a connection between criterion c and information sources s. The strength of connection $r = (c, s)$ is denoted by $u(r)$, which is expressed by a subjective term shown in Table 8.3. The connecting strength $u(r)$ will be used in the fuzzification method for objective information in next section.

8.2.5 A fuzzification method for objective information

Objective information (data) is often provided in numeric values. The FMP model uses fuzzy numbers to express objective information. Each data is transferred to a fuzzy number through a fuzzification method.

Suppose information source s is connecting to criterion c with connecting strength $u(r)$, then for any data v from s, let μ be a fuzzy number on the universal $U = [u_0, u_1]$ of v such that

(1) $\mu(x) = 1$ if $x = v$;

(2) $\mu(x) = 1$ for any $x \in U$ when $u(r) = 0$;

(3) $\mu(x) = 0$ if $x \neq v$ when $u(r) = 1$;

(4) $\mu(x) \geqslant \mu(y)$ if x is nearer to v than y.

The fuzzy number μ is called a fuzzy approximate of v. For example, the following function can be used in a fuzzification of v:

$$\mu(x) = \begin{cases} 0, & |x - v| \geqslant \dfrac{1}{\tan(\frac{\pi}{2}u(r))}; \\ 1 - \tan(\dfrac{\pi}{2}u(r))|x - v|, & |x - v| < \dfrac{1}{\tan(\frac{\pi}{2}u(r))}. \end{cases} \tag{8.4}$$

Because μ is not a fuzzy number on the interval [0, 1], the FMP model, then, maps μ to an element $\tilde{\mu}$ in $\mathscr{F}_{[0,1]}$ by

$$\tilde{\mu}(y) = \mu(x) \text{ if } y = \frac{x - u_0}{u_1 - u_0}, \forall x \in U, y \in [0, 1]. \tag{8.5}$$

The aim of the fuzzification method is to reduce fuzziness implied in the support information. As the issue of reducing uncertainty in collected information is a very difficult problem, this chapter only adopts above fuzzification method strategy and more work will be done in the future.

8.2.6 A fuzzy aggregation method

In the FMP model, the decision matrix (also denoted by T^i) is in the following from:

$$T^i = \begin{pmatrix} T_s & 0 \\ 0 & T_o \end{pmatrix}, \tag{8.6}$$

where T_s is a matrix for subjective assessments (e.g., the decision matrix in the model $\mathcal{M}$) and T_o is a matrix for objective data. Then the following fuzzy aggregation method is adapted to obtain an overall assessment on each alternative. The fuzzy aggregation method is conducted in two stages. In stage one, decision information for leaf criteria is aggregated to generate assessments on leaf criteria. In the second stage, assessments on leaf criteria are aggregated to generate assessments on non-leaf criteria.

The primary aggregation algorithm is illustrated below by taking a criterion c and its sub-criteria for example.

Suppose $\tilde{v}_1$, ..., $\tilde{v}_n$ is the assessments for sub-criteria c_1, ..., c_n of the criterion c. Let wc_1, ..., wc_n be the weights of c_1, ..., c_n, respectively. Here $\tilde{v}_j$ and wc_j, $j = 1, \ldots, n$, are fuzzy numbers on [0, 1]. Then the assessment on c by the fuzzy aggregation method is:

$$\tilde{v} = \sum_{j=1}^{n} \tilde{w}_j \tilde{v}_j, \tag{8.7}$$

and $\tilde{w}_j$ is the normalized weight for wc_j by

$$\tilde{w}_j = \frac{wc_j}{\sum_{j=1}^{n} (wc_j)_0^{\mathrm{R}}}, \quad j = 1, \ldots, n, \tag{8.8}$$

where $(wc_j)_0^{\mathrm{R}}$ is the right-end point of 0-cut of wc_j.

8.2.7 Fuzzy ranking of alternatives

To rank alternatives based on the FMP model, the following ranking method is used. Suppose $\tilde{v}^i$ is the assessments for alternatives a_i, $i = 1, \ldots, m$, and $\tilde{v}^i$ is a fuzzy number on the real interval [0, 1]. the FMP model defines two ideal assessments $\tilde{v}^+$ and $\tilde{v}^-$ on [0, 1] as follows:

$$\tilde{v}^+ = \begin{cases} 0, & x \neq 1; \\ 1, & x = 1. \end{cases}, \quad \tilde{v}^- = \begin{cases} 0, & x \neq 0; \\ 1, & x = 0. \end{cases} \tag{8.9}$$

Then the reference distance of $\tilde{v}^i$ to these two solutions is

$$d_i = \frac{1}{2}\left(d(\tilde{v}^i, \tilde{v}^-) + (1 - d(\tilde{v}^i, \tilde{v}^+))\right), \tag{8.10}$$

where $d(a, b)$ is a fuzzy distance of two fuzzy numbers a, b and defined as

$$d(\tilde{a}, \tilde{b}) = \left(\int_0^1 \frac{1}{2}\left[\left(\tilde{a}_\lambda^L - \tilde{b}_\lambda^L\right)^2 + \left(\tilde{a}_\lambda^R - \tilde{b}_\lambda^R\right)^2\right] \mathrm{d}\lambda\right)^{1/2}. \tag{8.11}$$

Based on the reference distance d_i, alternatives are ranked and the alternative with bigger value will be better.

8.2.8 Summary and an example

Summarizing above, the presented the FMP model is illustrated in eight steps as shown in Table 8.1.

Let us consider the following example. For convenience, suppose the settings are shown as Figure 8.2 and Table 8.2. In this example, criterion c_2 receives objective information from information source inf and criterion c_1 and its sub-criteria receive subjective information from evaluators. In the settings table, each fuzzy number $\tilde{v}$

Table 8.1: Outlines of the FMP model

Steps of the FMP model
(decision information input module in DECIDER)
Step 1: identify alternatives. **Step 2**: identify hierarchy of criteria and evaluators as well as their weights. **Step 3**: identify information sources and its connection with criteria. **Step 4**: collect information from information sources. **Step 5**: evaluators evaluate collected information to generate initial decision matrix for each alternative. *(decision information process module in DECIDER)* **Step 6**: apply the fuzzification method to assessments in initial decision matrix. **Step 7**: apply the fuzzy aggregation method to obtain overall assessment on each alternative. **Step 8**: generate ranking for each alternative by the fuzzy aggregation method and ranking strategy.

presents a triangular fuzzy number

$$\tilde{v} = \langle v - 0.05, v, v + 0.05 \rangle. \quad (8.12)$$

For instance, $\widetilde{0.7}$ is the triangular fuzzy number $\langle 0.65, 0.7, 0.75 \rangle$.

Figure 8.2: Example settings of a decision problem

By the summarized steps and the problem's settings, we illustrate the process from step 6. First, the objective data 0.06 and 0.08 are transferred to fuzzy numbers. Here, the following two fuzzy numbers, shown in Figure 8.3, are used:

$$\widetilde{0.06} = 1 - \frac{0.7|x - 0.06|}{0.94}, \quad \widetilde{0.89} = \max\{0, 1 - \frac{0.7|x - 0.89|}{0.11}\}. \quad (8.13)$$

Table 8.2: An example settings

criteria/evaluators	c_1	c_2	c_{11}	c_{12}	e_1	e_2	e_{21}	e_{22}	inf
weights/strength	$\widetilde{0.7}$	$\widetilde{0.6}$	$\widetilde{0.9}$	$\widetilde{0.1}$	$\widetilde{0.8}$	$\widetilde{0.2}$	$\widetilde{0.5}$	$\widetilde{0.7}$	0.3

for alternative a_1					for alternative a_2				
	e_1	e_{21}	e_{22}	inf		e_1	e_{21}	e_{22}	inf
c_{11}	$\widetilde{0.36}$	$\widetilde{0.96}$	$\widetilde{0.84}$		c_{11}	$\widetilde{0.78}$	$\widetilde{0.91}$	$\widetilde{0.37}$	
c_{12}	$\widetilde{0.14}$	$\widetilde{0.64}$	$\widetilde{0.26}$		c_{12}	$\widetilde{0.54}$	$\widetilde{0.63}$	$\widetilde{0.35}$	
c_2				0.06	c_{22}				0.89

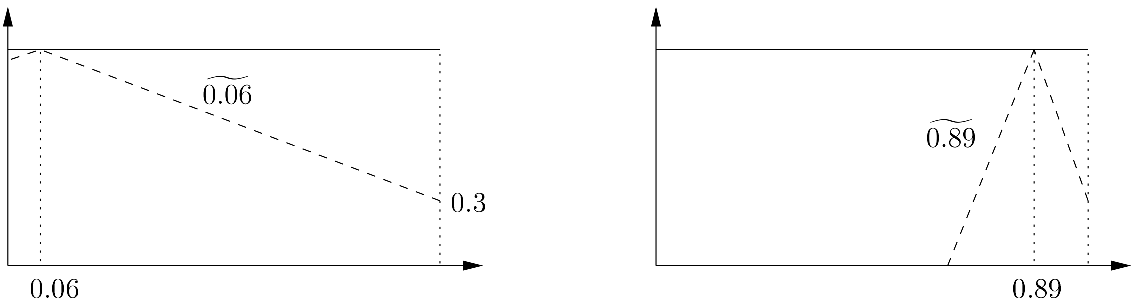


Figure 8.3: Example of converting objective data to subjective data

Next, we will aggregate the assessments for leaf criteria c_{11} and c_{12}. Take the criterion c_{11} for example. As e_{21} and e_{22} are real evaluators under e_2, the assessment of e_2 on c_{11} is obtained by

$$\widetilde{0.96} \cdot \frac{\widetilde{0.5}}{0.55 + 0.75} + \widetilde{0.84} \cdot \frac{\widetilde{0.7}}{0.55 + 0.75} \approx \widetilde{0.82}. \tag{8.14}$$ [2]

Then, the assessment of e_0 on c_{11} is calculated in a similar way and it is $\widetilde{0.41}$. For criterion c_{12}, the assessment is also calculated and is $\widetilde{0.11}$.

Next step, we will aggregate assessments on non leaf criteria c_1 and c_0. For c_1, we have

$$\widetilde{0.41} \cdot \frac{\widetilde{0.9}}{1.1} + \widetilde{0.11} \cdot \frac{\widetilde{0.1}}{1.1} \approx \widetilde{0.34}. \tag{8.15}$$

And for c_0, it is $\widetilde{0.20}$.

[2] In general, the aggregation result is not a triangular fuzzy number. In this example, the real result is a fuzzy set with 0-cuts 0.71 and 0.94.

Finally, a decision reference is calculated by (8.10) and it is 0.66. Similarly for alternative 2, the decision reference is 0.64. Hence alternative 1 is better than alternative 2.

8.3 DECIDER: A decision support system based on FMP model

Based on the presented FMP model, a decision support system, DECIDER, is designed and developed. This section will introduce the structure and functions of it. In next section, an application example is illustrated.

DECIDER is developed by using the Java$^{\text{TM}}$ programming language for running on platforms such as Linux and Windows. It is currently composed of four main modules (shown in Figure 8.4), i.e., decision information input module, decision information process modules, result comparison and analysis module, and result display module.

8.3.1 Decision information input module

Decision information input module provides interfaces for two kinds of settings relevant to an MCGDM problem. The first kind of settings includes information about criteria, evaluators, alternatives, information sources, and decision matrices, which are called basic information. The other kind of settings mainly involve in information transformation, such as fuzzification and operator selection, which are called assistant control.

Through designed basic information interfaces, users can set: 1) a multi-level hierarchy of evaluation criteria; 2) a list of alternatives; 3) a multi-level hierarchy of evaluation evaluators; 4) a list of information sources; and 5) a set of decision matrices. Criteria hierarchy settings mainly include criteria identity, weights, as well as

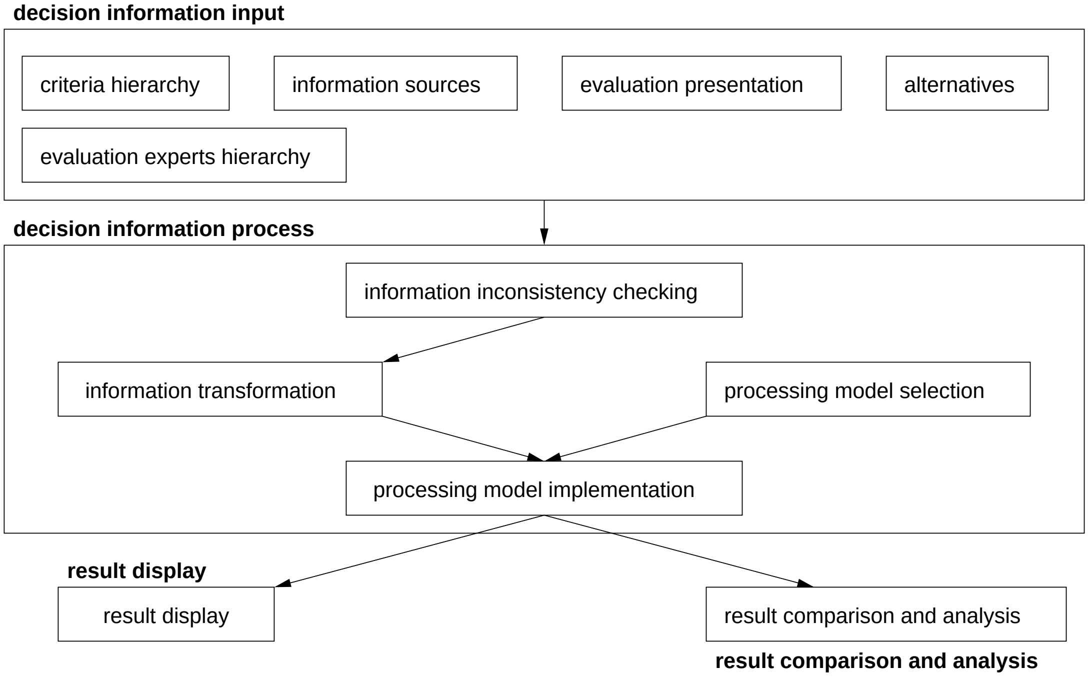


Figure 8.4: Main modules in DECIDER

assessment data types. Alternative settings include alternative identity and descriptions. Evaluators' hierarchy sets information about evaluators' organizations and their impacts (weights) on the decision problem. Information sources settings include information sources' reliabilities, information expression forms, and connections to criteria.

Assistant control includes comparison table between different evaluation term sets, selection of aggregation operators, parameters for distribution of evaluator terms, and parameters for fuzzification.

In real application, the used evaluation term set is not unique. DECIDER provides six sets of evaluation term sets, shown in Table 8.3, for information expressions. The six sets of evaluation terms mainly focus on three kinds of information presentation forms (data types), i.e., numeric values (e.g., numeric grades such as linguistic terms with integer labels 1, 2, ... and real numbers in given interval such as $0.5 \in [0, 1]$), Boolean values (e.g., *true* and *false*), and subjective terms in fuzzy numbers (e.g., linguistic terms such as *very high* and linguistic hedges such as *absolutely*). These three kinds of data types are used to express information such as criteria' weights, evaluators' weights, decision matrix, as well as strength of connections.

Table 8.3: Data types in DECIDER

data type	named expression	terms/values list
subjective terms	standard score	0, 1, . . ., 100
	linguistic weight	*Absolutely unimportant*, *Unimportant*, *Less important*, *Important*, *More important*, *Strongly important*, *Absolutely important*
	linguistic score	*Lowest*, *Very low*,*Low*, *Medium*, *High*, *Very high*, *Highest*
	numeric grade	1, 2, 3, 4, 5, 6, 7
objective values		user defined real interval
Boolean values		*true* and *false*

By default, all evaluation terms are symmetric distributed. However, users can assign the corresponding relationship between different terms through filling a comparison table. For example, user can assign "100" from "standard scale" to the term "absolutely important" in "linguistic weights." In order to integrate information in different expressions, DECIDER provide another method to illustrate corresponding relationship between two types of subjective terms except for directly assignment. The method takes the standard scale from 0 to 100 as the basis of comparison. A term in other three subjective expression sets is assigned an integer to be its label. For instance, user can assign 90 to the term *Absolutely important* and assign 100 to the term *Highest*. This integer indicates user's preference and understanding of that term. If a user cannot clearly assign labels to all terms, DECIDER can generate those labels based on user's expected distribution of them. Suppose $S = \{s_1, s_2, \ldots, s_n\}$ is a set of subjective terms and $s_1 < s_2 < \cdots < s_n$ gives a possible order of those terms. If the user's expecting distribution of terms in S is "equal ratio" with ratio γ ($\gamma > 0$), i.e., $s_{j+1} - s_j = r \cdot (s_j - s_{j-1})$, then DECIDER can generate an approximate label to each unassigned term based on assigned terms. For example, if a user assigns 0 to *Absolutely unimportant* and 90 to *Absolutely important* and expects the terms distribution is $\gamma = 1.0$, i.e., terms are equally distributed through 0 to 90. Then DECIDER will assign 15 to *Unimportant*, 30 to *Less important*, 45 to *Important*, 60 to *More important*, and 75 to *Strongly important*. If the user want to adjust the label for *Important* to 75, then DECIDER will adjust labels for terms *Unimportant*, *Less important*, *More important*, and *Strongly important* to 25, 50, 80, and 85, respectively. When the user expects $\gamma = 3.0$ and label for *Important* being 60, then DECIDER will approximately assign 5, 18, 62, and 69 to the reminder terms. Based on the adjustment of γ and assigned terms, a user preferred terms distribution is obtained.

For the numeric values, DECIDER allows a user to define the interval of value range and a preferred value. The interval indicates the range of meaningful values, such as an interval [-30, 30] for temperature of an area. In DECIDER, the interval's ends are mapped to 0 or 100 respectively based on interpretation of the preferred value and preferred order of values in it.

A preferred value has various interpretations in real applications. DECIDER treats preferred value in three interpretations, i.e., threshold value (T), medium value (M), and expected value (E). A threshold means an elementary requirement, for example, mark 60 is a basic requirement to pass a testing. An expected value is the most desired situation, for example, a product's cost under consideration of its quality. A medium value refers the average status of preference.

DECIDER defines four kinds of orders for labelling an interval:

O1 "the larger the better",

O2 "the smaller the better",

O3 "the nearer the better",

O4 "the farther the better".

Based on the interpretation of a preferred value and the four orders, six linear transformation methods are defined in DECIDER to label values in an interval.

In the following, let $[a, b]$ be the interval, and $p \in [a, b]$ be the preferred value. Then the label of $x \in [a, b]$ is given as follows:

Case 1: (T+O1) In this case, DECIDER assigns 0 to any $x \in [a, p]$, and for any $x \in [p, b]$ lets

$$l(x) = \lfloor \frac{x-p}{b-p} \cdot 100 \rfloor \tag{8.16}$$

be the label of x, where $\lfloor \cdot \rfloor$ is the floor function.

Case 2: (T+O2) In this case, DECIDER assign 0 to any $x \in [p, b]$, and for any $x \in [a, p]$ lets

$$l(x) = \lfloor \frac{p-x}{p-a} \cdot 100 \rfloor \tag{8.17}$$

be the label of x.

Case 3: (M+O1) In this case, DECIDER assigns 50 to p, and

$$l(x) = \begin{cases} \lfloor \frac{x-a}{p-a} \cdot 50 \rfloor, & x \in [a, p]; \\ \lfloor (\frac{x-p}{b-p} + 1) \cdot 50 \rfloor, & x \in [p, b]. \end{cases} \tag{8.18}$$

Case 4: (M+O2) In this case, DECIDER assigns 50 to p, and

$$l(x) = \begin{cases} \lfloor (\frac{p-x}{p-a} + 1) \cdot 50 \rfloor, & x \in [a, p]; \\ \lfloor \frac{b-x}{b-p} \cdot 50 \rfloor, & x \in [p, b]. \end{cases} \tag{8.19}$$

Case 5: (E+O3) In this case, DECIDER assigns 100 to p, and

$$l(x) = \begin{cases} \lfloor \frac{x-a}{p-a} \cdot 100 \rfloor, & x \in [a, p]; \\ \lfloor \frac{b-x}{b-p} \cdot 100 \rfloor, & x \in [p, b]. \end{cases} \tag{8.20}$$

Case 6: (E+O4) In this case, DECIDER assigns 100 to p, and

$$l(x) = \begin{cases} \lfloor(1 - \frac{x-a}{p-a}) \cdot 100\rfloor, & x \in [a,p]; \\ \lfloor(1 - \frac{b-x}{b-p}) \cdot 100\rfloor, & x \in [p,b]. \end{cases} \tag{8.21}$$

For Boolean values, DECIDER directly assigns 100 to *true* and 0 to *false*.

8.3.2 Model selection

As mentioned above, a core task of model $\mathcal{M}$ is selecting approximate aggregation operators $\circ$ and $\diamond$. Once those two operators are given, a specific processing model can be constructed accordingly. In DECIDER a module is developed to implement the functionality of selecting aggregation operators to construct different decision models. Users are allowed to select aggregation operators based on the natures of decision problem and criteria.

Moreover, DECIDER also provides a switch for using FMP model or model $\mathcal{M}$. The default setting of the switch is model $\mathcal{M}$. Users can select using FMP model if they concern about information sources. Otherwise, they can use the traditional model.

8.3.3 Decision information process

Once users have input necessary information and select aggregation operators, the processing model is fixed and the implementation of aggregation of assessments is trivial.

8.3.4 Result display, comparison and analysis

After execution of process, overall assessments on alternatives are displayed to user, as shown in Figure 8.6. Users are allowed to check assessments in terms of criteria and evaluators through their respective hierarchies. Moreover, users can also observe the changes through selecting different aggregation operators and modifying inputs.

Except for aforementioned modules, several other modules for DECIDERare designed and some parts of them are under study and development. Those modules include a module for imputation of missing assessments, a operator-base for aggregation process, a powerful comparison and analysis module to conduct sensitivity analysis, as well as a recommending module to provide operator selection suggestions.

8.4 A case study on new product development

DECIDER has been used and tested in several applications in the cooperation with France and Belgian colleagues. As DECIDER is still under updated, this section just illustrates a simple application in garment new product development evaluation. A more detailed process illustration is referred to Lu *et al.* (2008).

Garment new product development evaluation is a typical MCGDM problem, which involves not only subjective judgments but also objective sampling data. Under the concept of "well-being" in garment product design, a new product is evaluated based on a three-level hierarchy of criteria as shown in Figure 8.5. Based on the hierarchy of criteria, evaluator survey results are used to evaluate 10 new products.

In this example, the decision information includes subjective judgments expressed in linguistic terms and objective values in an interval. All these information are collected from reliable information sources and consulting evaluators. Hence, decision matrices are directly used in evaluation process and need not to be adjusted.

To implement the presented FMP model, we select the fuzzy aggregation method and ranking strategy described in Section 8.2 for primary aggregation. Meanwhile, we distribute all terms equally.

After input basic information and assistant control parameters, we obtain the final evaluation result shown in Figure 8.6. Consulting experts in this application are satisfied with the evaluation result.

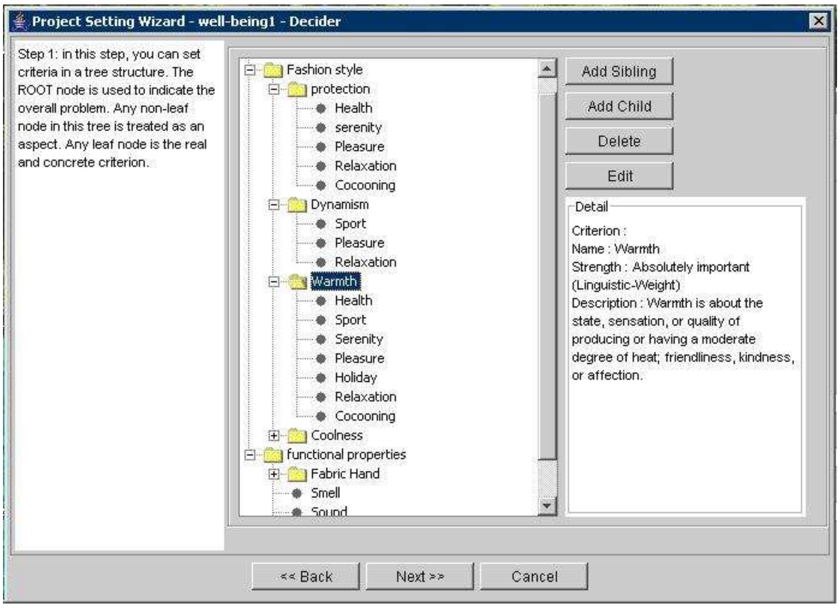


Figure 8.5: Well-being new garment products evaluation model

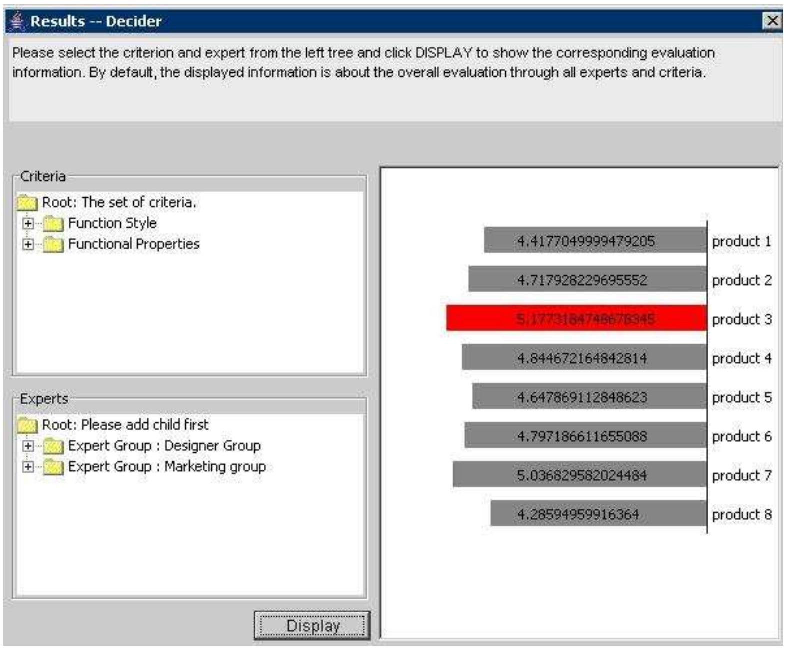


Figure 8.6: Final ranking result for new garment products under the well-being

8.5 Evaluations

We have conducted a survey about the developed functions of DECIDER. The survey response indicates that above 75 percents users are satisfied with DECIDER's user-friend interfaces, flexible settings for processing model, various information transfer methods, and reasonable results. At the same time, consulting users also provide some suggestions on the design for user interface and the presented FMP model.

Chapter 9 Conclusions and Future Works

This thesis studies four aspects of decision information processing in dynamic decision-making, i.e., information inconsistency detection, multi-source information integration, prediction with multiple periodic factor information, and decision information similarity measurement between decision makers. These four aspects are closely related to the problem of processing human factors in the development and deployment of a people-centred warning system.

Chapter 4 focuses on detecting potential data inconsistency and logical inconsistency in decision information. Section 4.2 develops a rule-map technique to organize multiple descriptive models of the functional pattern of a monitored object. A five-step RMDID method for the data inconsistencies detection problem is then proposed. By using the RMDID method, experiments have been conducted for two real data sets. Experiment results show that the RMDID method can take advantage of multiple descriptive models and, in particular, is effective even if prior knowledge about the monitored object is unavailable. Considering domain knowledge is dependent on the state of objects in that domain, Section 4.3 presents a state-based domain knowledge representation technique and proposes the SCLID method for detecting logical inconsistency in domain-specific real observations. In the state-based domain knowledge representation technique, a piece of domain knowledge is represented by state combinations of relevant objects. Thus, the logical relationship between pieces of knowledge is defined through those state combinations. Furthermore, the strict and partial consistency in the domain knowledge base is also defined by those state combinations. This knowledge representation technique has the flexibility to depict domain-specific knowledge. The developed SCLID method is an application of the state-based domain knowledge representation technique. It includes five main steps which are implemented through the coupling and extracting operations on state combinations. This implementation combines the merits of reasoning-based and graph-based inconsistency detection methods. To test and illustrate the efficiency of the SCLID method, two experiments are described. Experimental results indicate that the SCLID method can be used to detect the logical inconsistency of real-time observations and can also be used to detect the logical inconsistency of knowledge in knowledge bases. These features are very important in warning system developments and deployments.

Concerning the experimental results, our further study of this issue includes improving the performance of the rule-map technique and applying the RMDID method to warning system development. The rule-map technique is the basis of the proposed RMDID method and is designed to support dynamic decision-making. The performance of the rule-map technique is influenced by the performance of selected descriptive models. Since the nature of a dynamic decision-making situation will affect the performance of an established descriptive model, more research will be conducted to improve each model's performance and in turn enhance the performance of the rule-map technique. Note that there are various data representation forms in real-time information, and we need to understand their natures further and apply them to real applications.

Chapter 5 focuses on effectively integrating multi-source and qualitative (subjective) information. It first extends the decision hierarchy (the PM) to XPM which can improve the capability of illustrating domain knowledge and provide a facility for analysing information processing. The strengths of indicators are interpreted from a logical viewpoint. Finally, it proposes a QII method which has been used for nuclear safeguards information management. In the QII method, two aggregation operator selection strategies are proposed to optimize information integration processing. The main advantages of the QII method are that: 1) it employs transformation function and grouping aggregation to integrate multiple source information so that the information inconsistency issue is solved; and 2) it uses a logical inference technique to generate two aggregation strategies. Therefore, the two aggregation strategies are more consistent with people's experiences.

Case study indicates that two similar results are obtained by using different aggregation operators. This is obviously an expected result in a real situation because different decision makers may select and use different aggregation operators to generate their situation awareness. As a further study, we will improve the performance of the QII method on issues, including selecting appropriate applied implication operators and identifying qualitative information semantics.

All implication operators are rooted in a specific philosophical and logical background and are therefore more suited to some situations than others. Similarly, an aggregation operator has distinctive properties which are also domain specified. Moreover, the semantics of qualitative information involves not only a suitable representation form but also deep understanding of the background knowledge. Our future work will also give a detailed analysis of the specific natures of different implication operators, different aggregation operators, and knowledge representations that may contribute to establish an effective approach for integrating qualitative information.

Chapter 6 focuses on generating predictions from information affected by multiple periodic factors. This chapter presents a CFS-based prediction method, the PSAOP method, for solving MPFP problems. In the PSAOP method, past experience and knowledge about the periodically changed factors and the predicted event are firstly formalized by CFSs; observations about those factors are then represented by complex-valued membership degrees. A novel product-sum aggregation operator, the PSAO, is used to integrate data from these factors; finally, the prediction is made by searching the label for the product-sum of current data. The PSAOP method is demonstrated with its application to annual sunspot number prediction and bushfire danger rating prediction. Our study indicates that: 1) the PSAOP method is simple but has high performance and adaptability. It takes advantage of the CFS to describe semantic uncertainty and periodicity in information; 2) the PSAO can deal with the semantic uncertainty and periodicity in data from multiple periodically changing factors in MPFP problems simultaneously. In particular, the PSAO can integrate inputs with a quasi-order.

During the case studies, we observed that there are still important issues to be studied. Firstly, the presented PSAOP method closely depends on the time series prediction methods for each factor. How to effectively use existing data (in particular historic records) to build CFSs is still little known. This is an important theoretical and practical issue in CFS-related research. Secondly, when the observations of factors are qualitative expressions, the accuracy of some time series methods may be lost; hence, new methods for processing time series with qualitative data are required. We have noted relevant studies in the framework of fuzzy time series. These studies may be helpful for improving the performance of the presented PSAOP method. Thirdly, the phase part of each complex-valued membership degree is a unique feature of the CFS other than conventional fuzzy sets. However, how to rationally define and interpret this features is still a big question when applying CFSs to real applications. Although the phase definition is applicable in our experiments, developing a rational method to define the phase part is still a pressing issue because it may affect the performance of the presented method. We will pay more attention to this issue. Finally, the PSAOP method presented is a declarative method which is remarkably different to inductive learning prediction methods such as the ANCFIS method. Although we note that the performance can be improved by adjusting the weights of factors and selecting a more appropriate labelling strategy, these issues are not discussed in detail in this thesis and will be studied further in the future.

Chapter 7 focuses on measuring the similarity of decision information. To reduce the potential risk of putting an inappropriate decision into practice, measuring opinion similarity between participants is an important issue which has not been solved. To solve the MOSP problem, this chapter develops a gradual aggregation algorithm to model the dynamic generation of a decision and to process the missing value in it. Based on the gradual aggregation algorithm, a TLSM method for the MOSP problem is presented which measures the similarity between two opinions at the assessment level, the criterion level, and the decision level. Applying the TLSM method, two applications in social policy selection and energy policy evaluation are conducted.

The main contributions of this research are as follows. Firstly, the TLSM method provides a processing framework for the MOSP problem. The MOSP problem is a significant practical topic in many applications, although few research works have been undertaken in this area. Existing opinion similarity measuring methods can tackle a part of the MOSP problem, but they do not present a whole solution for it. Secondly, the small size of relevant opinion samples is a primary obstacle that prevents existing statistical learning techniques from being applied to the MOSP problem. The TLSM method overcomes them to some extent. Moreover, the TLSM method combines an opinion with its provider in its entire processing. This helps to develop more effective opinion similarity measurement and analysis techniques to overcome difficulties resulting from the separation of opinions and their providers in real applications. Finally, the experiments indicate that the TLSM method effectively handles missing data, unclear information, and linguistic assessments by adjusting the gradual aggregation algorithm. Highly satisfactory results have been obtained from the experiments.

Based on these experiments, some issues will be studied further. Firstly, the gradual aggregation algorithm is a technique to integrate information according to a group of inputs. The process order of the inputs has special meaning and impact on the final result. This study rearranges the inputs according to the descent order of the weights of criteria and a satisfactory result is obtained; nevertheless, the GAA's nature is still unclear and requires further study. Secondly, missing data and unclear answers are very common in real applications. The TSLM method treats them as distinct without distinguishing their real meanings and utilities. This is an intuitive and simple processing strategy. Whether there is a better strategy is a further area requiring investigation. Finally, the MOSP problem is a special case of the user opinion analysis and behaviour modelling problem. Due to variations in the natures of different application contexts, effective techniques for solving the user opinion analysis and behaviour modelling problem have not yet been found. Our next step is to extend the TLSM method and develop new techniques to provide applicable solutions for both the MOSP problem and the user opinion analysis and behaviour modelling problem.

Chapter 8 focuses on decision-making automation. This chapter first presents an FMP model for MCGDM in a decision context with multi-level criteria, multi-level evaluators and uncertain information with fuzziness. Moreover, the FMP model takes information sources into account in the information process. A decision support system, DECIDER, is developed to implement this FMP model. DECIDER can deal with both objective and subjective inputs at the same time by expressing them in fuzzy numbers and can aggregate all decision makers' judgments to form a higher-level satisfactory solution to the group of decision-makers. It has been tested and applied in real applications, including fabric material ranking, strategy evaluation, and nonwoven product assessment.

References

Adomavicius, G. and Tuzhilin, A. 2005, ‘Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions’, *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 737–749.

Aghakhani, S. and Dick, S. 2010, ‘An on-line learning algorithm for complex fuzzy logic’, in *IEEE International Conference on Fuzzy Systems (FUZZ)*, pp. 1–7.

Ahn, B. S., Park, K. S., Han, C. H. and Kim, J. K. 2000, ‘Multi-attribute decision aid under incomplete information and hierarchical structure’, *European Journal of Operational Research*, vol. 125, pp. 431–439.

Amgoud, L. and Kaci, S. 2007, ‘An argumentation framework for merging conflicting knowledge bases’, *International Journal of Approximate Reasoning*, vol. 45, pp. 321–340.

Balopoulos, V., Hatzimichailidis, A. G. and Papadopoulos, B. K. 2007, ‘Distance and similarity measures for fuzzy operators’, *Information Sciences*, vol. 177, pp. 2336–2348.

Banisch, S. and Araújo, T. 2010, ‘On the empirical relevance of the transient in opinion models’, *Physics Letters A*, vol. 374, pp. 3197–3200.

Basher, R. 2006, ‘Global early warning systems for natural hazards: systematic and people-centred’, *Philosophical Transactions of the Royal Society A*, vol. 364, pp. 2167–2182.

Bass, T. 2000, ‘Intrusion detection systems and multisensor data fusion’, *Communications of ACM*, vol. 43, no. 4, pp. 99–105.

Beliakov, G. and Calvo, T. 2008, 'Interpolatory type construction of general aggregation operators', in H. Bustince, F. Herrera and J. Montero (eds), *Fuzzy Sets and Their Extensions: Representation, Aggregation and Models*, Springer, pp. 99–120.

Beliakov, G. and Warren, J. 2001, 'Appropriate choice of aggregation operators in fuzzy decision support systems', *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 6, pp. 773–784.

Bichescu, B. C. and Fry, M. J. 2009, 'A numerical analysis of supply chain performance under split decision rights', *Omega*, vol. 37, pp. 358–379.

Bosteels, K. and Kerre, E. E. 2007, 'A triparametric family of cardinality-based fuzzy similarity measures', *Fuzzy Sets and Systems*, vol. 158, pp. 2466–2479.

Botten, N. 1992, 'Complex knowledge-base verification using matrices', *Lecture Notes in Artificial Intelligence*, vol. 604, pp. 225–235.

Bruni, R. 2004, 'Discrete models for data imputation', *Discrete Applied Mathematics*, vol. 144, pp. 59–69.

Buckley, J. J. 1989, 'Fuzzy complex numbers', *Fuzzy Sets and Systems*, vol. 33, pp. 333–345.

Buckley, J. J. 1992, 'Fuzzy complex analysis II: Integration', *Fuzzy Sets and Systems*, vol. 49, no. 2, pp. 171–179.

Buckley, J. J. and Qu, Y. 1991, 'Fuzzy complex analysis I: Definition', *Fuzzy Sets and Systems*, vol. 41, no. 2, pp. 269–284.

Bustince, H., Herrera, F. and Montero, J. (eds) 2008, *Fuzzy Sets and Their Extensions: Representation, Aggregation and Models*, *Studies in Fuzziness and Soft Computing*, vol. 220, Springer-Verlag, Berlin Heidelberg.

Calvo, T., Kolesárová, A., Komorníková, M. and Mesiar, R. 2002a, *Aggregation Operators: Properties, Classes and Construction Methods*, Physica-Verlag: Springer, Heidelberg, Germany, Germany, chap. 1, pp. 3–104.

Calvo, T., Mayor, G. and Mesiar, R. 2002b, *Aggregation Operators: New Trends and Applications*, no. 97 in Studies in Fuzziness and Soft Computing, Physica-Verlag: Springer, Heidelberg.

Castro, J. L. and Zurita, J. M. 1998, 'A heuristic in rules based systems for searching of inconsistencies', *Information Sciences*, vol. 108, pp. 135–148.

Chakraborty, C. and Chakraborty, D. 2010, 'A fuzzy clustering methodology for linguistic opinions in group decision making', *Applied Soft Computting*, vol. 7, pp. 858–869.

Chang, C. L. and Lee, R. C. T. 1973, *Symbolic and Mechanical Theorem Proving*, Academic Press, Inc., Orlando, FL, USA.

Chen, H. and Zimbra, D. 2010, 'AI and opinion mining', *IEEE Intelligent Systems*, vol. 26, pp. 74–76.

Chen, S.-M. and Hwang, J.-R. 2000, 'Temperature prediction using fuzzy time series', *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 30, no. 2, pp. 263–275.

Chen, Z., Aghakhani, S., Man, J. and Dick, S. 2011, 'ANCFIS: A neuro-fuzzy architecture employing complex fuzzy sets', *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 2, pp. 305–322.

Cross, V. V. and Sudkamp, T. A. 2002, *Similarity and Compatibility in Fuzzy Set Theory*, Physica-Verlag, Heidelberg.

Das, G., Lin, K.-I., Mannila, K., Renganathan, G. and Smyth, P. 1998, 'Rule discovery from time series', in *Proc. of the Fourth International Conference on Knowledge Discovery and Data Mining*, pp. 16–22, URL `http://citeseer.ist.psu.edu/das98rule.html`.

Dawyndt, P., Vancanneyt, M., Meyer, H. D. and Swings, J. 2005, 'Knowledge accumulation and resolution of data inconsistencies during the integration of microbial information sources', *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 8, pp. 1111–1126.

de Amo, S. and Pais, M. S. 2007, 'A paraconsistent logic programming approach for querying inconsistent databases', *International Journal of Approximate Reasoning*, vol. 46, pp. 366–386.

De Baets, B., De Meyer, H. and Naessens, H. 2001, 'A class of rational cardinality-based similarity measures', *Journal of Computational and Applied Mathematics*, vol. 132, pp. 51–69.

De Baets, B., Janssens, S. and De Meyer, H. 2009, 'On the transitivity of a parametric family of cardinality-based similarity measures', *International Journal of Approximate Reasoning*, vol. 50, pp. 104–116.

De Gooijer, J. G. and Hyndman, R. J. 2006, '25 years of time series forecasting', *International Journal of Forecasting*, vol. 22, pp. 443–473.

Delgado, M., Herrera, F. and Herrera-Viedma, E. 2001, 'Combining linguistic information in a distributed intelligent agent model for information gathering on the internet', in P. Wang (ed), *Computing with Words*, John Wiley and Sons, pp. 251–276.

Delgado, M., Herrera, F., Herrera-Viedma, E. and Martínez, L. 1998, 'Combining numerical and linguistic information in group decision making', *Information Sciences*, vol. 107, pp. 177–194.

Deshmukh, A. Y., Bavaskar, A. B., Bajaj, R. R. and Keskar, A. G. 2008, 'Implementation of complex fuzzy logic modules with VLSI approach', *International Journal of Computer Science and Network Security*, vol. 8, no. 9, pp. 172–178.

Dick, S. 2005, 'Toward complex fuzzy logic', *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 3, pp. 405–414.

Emergency Management Australia 2010, 'EMA Disasters Database', URL `http:`

//www.ag.gov.au/ema/emaDisasters.nsf.

Esteban, J., Starr, A., Willetts, R., Hannah, P. and Bryanston-Cross, P. 2005, 'A review of data fusion models and architectures: towards engineering guidelines', *Neural Computing & Applications*, vol. 14, pp. 273–281.

Ferreira, R. J. and de Almeida, A. T. 2009, 'A multi-criteria decision model to determine inspection intervals of condition monitoring based on delay time analysis', *Reliability Engineering and System Safety*, vol. 94, pp. 905–912.

Finkele, K., Mills, G. A., Beard, G. and Jones, D. A. 2006, 'National daily gridded soil moisture deficit and drought factors for use in prediction of forest fire danger index in Australia', *Tech. Rep. 119*, Bureau of Meteorology Research Centre.

Fontini, F., Umgiesser, G. and Vergano, L. 2010, 'The role of ambiguity in the evaluation of the net benefits of the MOSE system in the Venice lagoon', *Ecological Economics*, vol. 69, pp. 1964–1972.

Franses, P. H. and Paap, R. 2004, *Periodic Time Series Models*, Oxford University Press, New York, USA.

Ganesan, K., Zhai, C. and Han, J. 2010, 'Opinosis: a graph based approach to abstractive summarization of highly redundant opinions', in *Proceedings of the 23rd International Conference on Computational Linguistics*, Beijing, China, pp. 340–348.

Gentil, S., Montmain, J. and Combastel, C. 2004, 'Combining FDI and AI approaches within causal-model-based diagnosis', *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 34, no. 5, pp. 2207–2221.

Graves, D. and Pedrycz, W. 2009, 'Fuzzy prediction architecture using recurrent neural networks', *Neurocomputing*, vol. 72, pp. 1668–1678.

Hájek, P., Havránek, T. and Jiroušek, R. 1992, *Uncertain Information Processing in Expert Systems*, CRC Press, Boca Raton, FL.

Hakanen, J., Miettinen, K. and Sahlstedt, K. 2011, 'Wastewater treatment: new insight provided by interactive multiobjective optimization', *Decision Support Systems*, vol. 51, pp. 328–337.

Hall, D. L. and Llinas, J. 1997, 'An introduction to multisensor data fusion', *Proceedings of the IEEE*, vol. 87, no. 1, pp. 6–23.

Hamilton, J. D. 1994, *Time Series Analysis*, Princeton University Press, Princeton, New Jersey.

Herrera, F., Herrera-Viedma, E. and Chiclana, F. 2001, 'Multiperson decision–making based on multiplicative preference relations', *European J. Operational Research*, vol. 129, pp. 372–385.

Herrera, F., Lopez, E. and Rodriguez, M. 2002, 'A linguistic decision model for promotion mix management solved with genetic algorithms', *Fuzzy Sets and Systems*, vol. 131, pp. 47–61.

Herrera, F. and Martínez, L. 2000, 'A 2-tuple fuzzy linguistic representation model for computing with words', *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 6, pp. 746–752.

Herrera-Viedma, E., Chiclana, F., Herrera, F. and Alonso, S. 2007, 'Group decision-making model with incomplete fuzzy preference relations based on additive consistency', *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 37, no. 1, pp. 176–189.

Herrera-Viedma, E., Herrera, F., Martínez, L., Herrera, J. C. and López, A. G. 2004, 'Incorporating filtering techniques in a fuzzy linguistic multi-agent model for information gathering on the web', *Fuzzy Sets and Systems*, vol. 148, pp. 61–83.

Ho, N. C. and Nam, H. V. 2002, 'An algebraic approach to linguistic hedges in Zadeh's fuzzy logic', *Fuzzy Sets and Systems*, vol. 129, pp. 229–254.

Ho, N. C., Nam, H. V., Khang, T. and Chau, N. 1999, 'Hedge algebras, linguistic–

valued logic and their application to fuzzy reasoning', *International Journal of Uncertainty Fuzziness Knowledge–Based Systems*, vol. 7, no. 4, pp. 347–361.

Ho, N. C. and Wechler, W. 1990, 'Hedge algebras: an algebraic approach to structure of sets of linguistic truth values', *Fuzzy Sets and Systems*, vol. 35, pp. 281–293.

Ho, N. C. and Wechler, W. 1992, 'Extended hedge algebras and their application to fuzzy logic', *Fuzzy Sets and Systems*, vol. 52, pp. 259–281.

Hu, M. and Liu, B. 2004, 'Mining and summarizing customer reviews', in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD 2004, ACM, New York, NY, USA, pp. 168–177.

Huang, B., Kechadi, T.-M., B.Buckley, Kiernan, G., Keogh, E. and Rashid, T. 2010, 'A new feature set with new window techniques for customer churn prediction in land-line telecommunications', *Expert Systems with Applications*, vol. 37, pp. 3657–3665.

Huang, W., Zhao, Y., Yang, S. and Lu, Y. 2008, 'Analysis of the user behavior and opinion classification based on the BBS', *Applied Mathematics and Computation*, vol. 205, pp. 668–676.

Hunter, A. 1998, 'Paraconsistent logics', in D. Gabbay and P. Smets (eds), *Handbook of Defeasible Reasoning and Uncertain Information*, Kluwer Academic Publishers, Dordrecht/Boston/London, pp. 13–43.

Hunter, A. 2003, 'Evaluating the significance of inconsistencies', in *Proceedings of the 2003 International Joint Conference on AI (IJCAI03)*, pp. 468–473.

Huynh, V.-N. and Nakamori, Y. 2005, 'A satisfactory-oriented approach to multiexpert decision-making with linguistic assessments', *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 35, no. 2, pp. 184–196.

Hyams, D. G. 2007, 'CurveExpert', URL `http://curveexpert.webhop.biz`.

Hyndman, R. J. 2006, 'Time series data library', Accessed in May 2006, URL `http:`

//www-personal.buseco.monash.edu.au/$\sim$hyndman/ TSDL/.

Ibargüengoytia, P. H. 2001, 'Real time intelligent sensor validation', *IEEE Transactions on Power Systems*, vol. 16, no. 4, pp. 770–775.

Inglada, J. and Mercier, G. 2007, 'A new statistical similarity measure for change detection in multitemporal SAR images and its extension to multiscale change analysis', *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 5, pp. 1432–1445.

Iosif, E. and Potamianos, A. 2010, 'Unsupervised semantic similarity computation between terms using web documents', *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 11, pp. 1637–1647.

Karsak, E. E. and Tolga, E. 2001, 'Fuzzy multi-criteria decision-making procedure for evaluating advanced manufacturing system investments', *International Journal of Production Economics*, vol. 69, pp. 49–64.

Keyno, H. S., Ghaderi, F., Azade, A. and Razmi, J. 2009, 'Forecasting electricity consumption by clustering data in order to decrease the periodic variable's effects and by simplifying the pattern', *Energy Conversion and Management*, vol. 50, pp. 829–836.

Kim, S. H. and Ahn, B. S. 1999, 'Interactive grouping decision making procedure under incomplete information', *European Journal of Operational Research*, vol. 116, pp. 498–507.

Kirchgässner, G. and Wolters, J. 2007, *Introduction to Modern Time Series Analysis*, Springer-Verlag, Berlin Heidelberg.

Klir, G. J. and Yuan, B. 1995, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice-Hall, NJ, USA.

Kou, G., Shi, Y. and Wang, S. 2011, 'Multiple criteria decision making and decision support system – guest editor's introduction', *Decision Support Systems*, vol. 51, no. 2.

Kuchar, J. K. and Yang, L. C. 2000, 'A review of conflict detection and resolution modeling methods', *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 179–189.

Kumar, M., Grag, D. P. and Zachery, R. A. 2006, 'A generalized approach for inconsistency detection in data fusion from multiple sensors', in *Proceedings of the 2006 American Control Conference*, Minneapolis, Minnesota, USA, pp. 2078–2083.

Larsen, H. L. and Yager, R. R. 2000, 'A framework for fuzzy recognition technology', *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, vol. 30, no. 1, pp. 65–76.

Lawry, J. 2001, 'A methodology for computing with words', *International Journal of Approximate Reasoning*, vol. 28, pp. 51–89.

Lawry, J. 2004, 'A framework for linguistic modelling', *Artificial Intelligence*, vol. 155, pp. 1–39.

Lawry, J. 2008, 'An overview of computing with words using label semantics', in H. Bustince, F. Herrera and J. Montero (eds), *Fuzzy Sets and Their Extensions: Representation, Aggregation and Models*, Springer-Verlag, Berlin Heidelberg, pp. 65–87.

Lee, L.-W., Wang, L.-H., Chen, S.-M. and Leu, Y.-H. 2006, 'Handling forecasting problems based on two-factors high-order fuzzy time series', *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 3, pp. 468–477.

Li, C. and Chan, F. 2011, 'Complex-fuzzy adaptive image restoration: an artificial-bee-colony-based learning approach', in *Proceedings of the Third International Conference on Intelligent Information and Database Systems*, ACIIDS'11, Springer-Verlag, Berlin, Heidelberg, pp. 90–99.

Li, C. and Chiang, T.-W. 2011, 'Complex fuzzy computing to time series prediction: a multi-swarm PSO learning approach', in *Proceedings of the Third International Conference on Intelligent Information and Database Systems*, ACIIDS'11, Springer-Verlag, Berlin, Heidelberg, pp. 242–251.

Lindell, M. K. 2000, 'An overview of protective action decision-making for a nuclear power plant emergency', *Journal of Hazardous Materials*, vol. 75, no. 2-3, pp. 113–129.

Liu, B. 2010, 'Sentiment analysis: a multifaceted problem', *IEEE Intelligent Systems*, vol. 25, no. 3, pp. 76–80.

Liu, J., Ruan, D. and Carchon, R. 2002, 'Synthesis and evaluation analysis of the indicator information in nuclear safeguards applications by computing with words', *International Journal of Applied Mathematics and Computer Science*, vol. 12, no. 3, pp. 449–462.

Liu, Z. and Morsy, S. 2007, 'Development of the physical model', *Tech. Rep. IAEA-SM-367/13/07*, IAEA.

Llinas, J., Bowman, C., Rogova, G., Steinberg, A., Waltz, E. and White, F. 2004, 'Revisiting the JDL data fusion model II', in *The 7th International Conference on Information Fusion*, vol. 2, vol. 2, pp. 1218–1230.

Lu, J., Ma, J., Zhang, G., Zhu, Y., Zeng, X. and Koehl, L. 2011, 'Theme-based comprehensive evaluation in new product development using fuzzy hierarchical criteria group decision-making method', *IEEE Transactions on Industrial Electronics*, vol. 58, no. 6, pp. 2236–2246.

Lu, J., Zhu, Y., Zeng, X., Koehl, L., Ma, J. and Zhang, G. 2008, 'A fuzzy decision support system for garment new product development', in *AI08: Proceedings of the 21st Australasian Joint Conference on Artificial Intelligence*, Springer-Verlag, Berlin, Heidelberg, pp. 532–543.

Lucas, C. 2010, 'On developing a historical fire weather data–set for Australia', *Australian Meteorological and Oceanographic Journal*, vol. 60, pp. 1–14.

Ma, J., Xu, Y., Ruan, D. and Zhang, G. 2007, 'A fuzzy-set approach to treat determinacy and consistency of linguistic terms in multi-criteria decision making', *International Journal of Approximate Reasoning*, vol. 44, no. 2, pp. 165–181.

Marichal, J.-L. 1998, *Aggregation Operations for Multicriteria Decision Aid*, Ph.D. thesis, Faculty of Sciences, University of Liége, Belgium.

Marichal, J.-L. 2000, 'An axiomatic approach of the discrete Choquet integral as a tool to aggregate interacting criteria', *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 6, pp. 800–807.

Marichal, J.-L. 2001, 'An axiomatic approach of the discrete Sugeno integral as a tool to aggregate interacting criteria in a qualitative framework', *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 1, pp. 164–172.

Maschio, I. 2007, 'A decision-support system for safeguards information analysis', *International Journal of Nuclear Knowledge Management*, vol. 2, no. 4, pp. 410–421.

Mata, F., Martínez, L. and Herrera-Viedma, E. 2009, 'An adaptive consensus support model for group decision-making problems in a multigranular fuzzy linguistic context', *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 2, pp. 279–290.

Maurya, M. R., Rengaswamy, R. and Venkatasubramanian, V. 2007, 'Fault diagnosis using dynamic trend analysis: a review and recent developments', *Engineering Applications of Artificial Intelligence*, vol. 20, pp. 133–146.

Mazure, B., Saïs, L. and Grégoire, E. 1997, 'Checking several forms of consistency in nonmonotonic knowledge-bases', in D. M. Gabbay, R. Kruse, A. Nonnengard and H. J. Ohlbach (eds), *Qualitative and Quantitative Practical Reasoning, First International Joint Conference on Qualitative and Quantitative Practical Reason-*

ing ECSQARU-FAPR'97, Bad Honnef, Germany, *Lecture Notes in Computer Sciences*, vol. 1244, Springer-Verlag: Berlin Heidelberg, *Lecture Notes in Computer Sciences*, vol. 1244, pp. 122–130.

Mendel, J. M. 2002, 'An architecure for making judgements using computing with words', *International Journal of Applied Mathematics and Computer Science*, vol. 12, no. 3, pp. 325–335.

Mendel, J. M. 2007a, 'Computing with words and its relationships with fuzzistics', *Information Sciences*, vol. 177, pp. 988–1006.

Mendel, J. M. 2007b, 'Computing with words: Zadeh, Turing, Popper and Occam', *IEEE Computational Intelligence Magazine*, vol. 2, pp. 10–17.

Mendel, J. M. 2007c, 'Type-2 fuzzy sets and systems: an overview', *IEEE Computational Intelligence Magazine*, vol. 2, pp. 20–29.

Mesiar, R., Kolesárová, A., Calvo, T. and Komorníková, M. 2008a, 'A review of aggregation functions', in H. Bustince, F. Herrera and J. Montero (eds), *Fuzzy Sets and Their Extensions: Representation, Aggregation and Models*, *Studies in Fuzziness and Soft Computing*, vol. 220, Springer-Verlag: Berlin Heidelberg, pp. 121–144.

Mesiar, R., Špirková, J. and Vavríková, L. 2008b, 'Weighted aggregation operators based on minimization', *Information Sciences*, vol. 178, pp. 1133–1140.

Meyer, P. and Roubens, M. 2005, 'On the use of the Choquet integral with fuzzy numbers in multiple criteria decision aiding', in *Proceedings of EUSFLAT-LFA 2005 conference*, pp. 210–215.

Meyer, P. and Roubens, M. 2006, 'On the use of the Choquet integral with fuzzy numbers in multiple criteria decision support', *Fuzzy Sets and Systems*, vol. 157, pp. 927–938.

Mileti, D. S. and Sorensen, J. H. 1990, 'Communication of emergency public warnings: a social science perspective and state-of-the-art assessment', *Tech. Rep. 6609*, Oak Ridge National Laboratory, U.S. Department of Energy, Oak Ridge, TN, USA.

Miller, D. T. and Morrison, K. R. 2009, 'Expressing deviant opinions: believing you are in the majority helps', *Journal of Experimental Social Psychology*, vol. 45, pp. 740–747.

Mitchell, H. B. 2007, *Multi-Sensor Data Fusion: An Introduction*, Springer-Verlag: Berlin Heidelberg.

Moses, D., Degani, O., Teodorescu, H.-N., Friedman, M. and Kandel, A. 1999, 'Linguistic coordinate transformations for complex fuzzy sets', in *Proceedings of 1999 IEEE International Fuzzy Systems Conferences*, vol. 3, Seoul, Korea, vol. 3, pp. 1340–1345.

Motro, A. and Anokhin, P. 2006, 'Fusionplex: resolution of data inconsistencies in the integration of heterogeneous information sources', *Information Fusion*, vol. 7, pp. 176–196.

Mues, C. and Vanthienen, J. 2004a, 'Efficient rule base verification using binary decision diagrams', in *Proc. of Database and Expert Systems Application*, *Lecture Notes in Computer Science*, vol. 3180, *Lecture Notes in Computer Science*, vol. 3180, pp. 445–454.

Mues, C. and Vanthienen, J. 2004b, 'Improving the scalability of rule base verification using binary decision diagrams: an empirical study', in *Proc. of Advances in Artificial Intelligence*, *Lecture Notes in Computer Science*, vol. 3238, *Lecture Notes in Computer Science*, vol. 3238, pp. 381–395.

Munda, G. 2008, *Social Multi-Criteria Evaluation for a Sustainable Economy*, Springer-Verlag: Berlin Heidelberg.

Munda, G. 2009, 'A conflict analysis approach for illuminating distributional issues in sustainability policy', *European Journal of Operational Research*, vol. 194, no. 1, pp. 307–322.

Murray, T. J. and Tanniru, M. R. 1991, 'Control of inconsistency and redundancy in PROLOG-type knowledge base', *Expert Systems with Applications*, vol. 2, no. 4, pp. 321–331.

Nanopoulos, A., Rafailidis, D., Symeonidis, P. and Manolopoulos, Y. 2010, 'MusicBox: personalized music recommendation based on cubic analysis of social tags', *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 2, pp. 407–412.

Németh, E., Lakner, R., Hangos, K. and Cameron, I. 2007, 'Prediction-based diagnosis and loss prevention using qualitative multi-scale models', *Information Sciences*, vol. 177, pp. 1916–1930.

Nguyen, H. T., Kandel, A. and Kreinovich, V. 2000, 'Complex fuzzy sets: towards new foundations', in *Proceedings of IEEE International Conference on Fuzzy Systems*, San Antonio, USA, pp. 1045–1048.

Nguyen, H. T., Kreinovich, V. and Shekhter, V. 1998, 'On the possibility of using complex values in fuzzy logic for representing inconsistencies', *International Journal of Intelligent Systems*, vol. 13, pp. 683–714.

Nguyen, N. T. 2005, 'Processing inconsistency of knowledge on semantic level', *Journal of Universal Computer Science*, vol. 11, no. 2, pp. 285–302.

Pandelaere, M., Briers, B., Dewitte, S. and Warlop, L. 2010, 'Better think before agreeing twice mere agreement: a similarity-based persuasion mechanism', *International Journal of Research in Marketing*, vol. 27, pp. 133–141.

Pang, B. and Lee, L. 2008, 'Opinion mining and sentiment analysis', *Foundations and Trend in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135.

Park, J. H. and Seong, P. H. 2002, 'An integrated knowledge base development tool for knowledge acquisition and verification for NPP dynamic alarm processing systems', *Annals of Nuclear Energy*, vol. 29, no. 4, pp. 447–463.

Pasi, G. and Yager, R. R. 2006, 'Modeling the concept of majority opinion in group decision making', *Information Sciences*, vol. 176, pp. 390–414.

Polat, F. 1993, 'UVT: A unification-based toold for knowledge base verification', *IEEE Expert–Intelligent Systems & Their Applications*, vol. 8, no. 3, pp. 69–75.

Preece, A. D. and Shinghal, R. 1994, 'Foundation and application of knowledge base verification', *International Journal of Intelligent Systems*, vol. 9, no. 8, pp. 683–701.

Qiu, D., Shu, L. and Mo, Z. 2009, 'Notes on fuzzy complex analysis', *Fuzzy Sets and Systems*, vol. 160, pp. 1578–1589.

Ramot, D., Friedman, M., Langholz, G. and Kandel, A. 2003, 'Complex fuzzy logic', *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 4, pp. 450–461.

Ramot, D., Milo, R., Friedman, M. and Kandel, A. 2002, 'Complex fuzzy sets', *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 171–186.

Ruan, D., Liu, J. and Carchon, R. 2003, 'Linguistic assessment approach for managing nuclear safeguards indicator information', *Logistic Information Management*, vol. 16, no. 6, pp. 401–419.

Russell, S. J. and Norvig, P. 1995, *Artificial Intelligence: A Modern Approach*, Prentice Hall, Upper Saddle River, NJ, 1st edn.

Scarpelli, H. and Gomide, F. 1994, 'A high level net approach for discovering potential inconsistencies in fuzzy knowledge bases', *Fuzzy Sets and Systems*, vol. 64, pp. 175–193.

Scotney, B. and McClean, S. 2003, 'Database aggregation of imprecise and uncertain evidence', *Information Sciences*, vol. 155, pp. 245–263.

Shang, H. L. and Hyndman, R. J. 2011, 'Nonparametric time series forecasting with dynamic updating', *Mathematics and Computers in Simulation*, vol. 81, pp. 1310–1324.

Sigall, H. and Aronson, E. 1967, ‘Opinion change and the gain-loss model of interpersonal attraction’, *Journal of Experimental Social Psychology*, vol. 3, pp. 178–188.

Song, Q. and Chisson, B. S. 1993, ‘Fuzzy time series and its models’, *Fuzzy Sets and Systems*, vol. 54, no. 3, pp. 269–277.

Stach, W., Kurgan, L. A. and Pedrycz, W. 2008, ‘Numerical and linguistic prediction of time series with the use of fuzzy cognitive maps’, *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 1, pp. 61–72.

Steinberg, A. N., Bowman, C. L. and White, F. E. 1999, ‘Revisions to the JDL data fusion model’, in B. V. Dasarathy (ed), *Sensor Fusion: Architectures, Algorithms, and Applications III*, vol. 3719, SPIE, vol. 3719, pp. 430–441.

Symeonidis, P., Nanopoulos, A. and Manolopoulos, Y. 2010, ‘A unified framework for providing recommendations in social tagging systems based on ternary semantic analysis’, *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 2, pp. 179–192.

Tamir, D. E., Jin, L. and Kandel, A. 2011, ‘A new interpretation of complex membership grade’, *International Journal of Intelligent Systems*, vol. 26, pp. 285–312.

Tan, S., Lim, C. and Rao, M. 2007, ‘A hybrid neural network model for rule generation and its application to process fault detection and diagnosis’, *Engineering Applications of Artificial Intelligence*, vol. 20, pp. 203–213.

The United Nations 2006, ‘Global Survey of Early Warning Systems’, .

Triantaphyllou, E. 2000, *Multi-Criteria Decision Making Methods: A Comparative Study*, Kluwer Academic Publishers, Dordrecht/Boston/London.

Ustun, O. and Demirtas, E. A. 2008a, ‘An integrated multi-objective decision-making process for multi-period lot-sizing with supplier selection’, *Omega*, vol. 36, pp. 509–521.

Ustun, O. and Demirtas, E. A. 2008b, ‘Multi-period lot-sizing with supplier selection using achievement scalarizing functions', *Computers & Industrial Engineering*, vol. 54, pp. 918–931.

Vechtomova, O. 2010, 'Facet-based opinion retrieval from blogs', *Information Processing and Management*, vol. 46, pp. 71–88.

Venkatasubramanian, V., Rengaswamy, R. and Kavuri, S. N. 2003a, 'A review of process fault detection and diagnosis, Part II: Qualitative models and search strategies', *Computers and Chemical Engineering*, vol. 27, pp. 311–326.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N. and Yin, K. 2003b, 'A review of process fault detection and diagnosis, Part III: Process history based methods', *Computers and Chemical Engineering*, vol. 27, pp. 327–346.

Venkatasubramanian, V., Rengaswamy, R., Yin, K. and Kavuri, S. N. 2003c, 'A review of process fault detection and diagnosis, Part I: Quantitative model-based methods', *Computers and Chemical Engineering*, vol. 27, pp. 293–311.

Vojnović, M., Cruise, J., Gunawardena, D. and Marbach, P. 2009, 'Ranking and suggesting popular items', *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 8, pp. 1133–1146.

Wang, X., De Baets, B. and Kerre, E. E. 1995, 'A comparative study of similarity measures', *Fuzzy Sets and Systems*, vol. 73, pp. 259–268.

Wu, P. and Su, S. Y. W. 1993, 'Rule validation based on logical deduction', in *Proc. 2nd International Conference on Information and Knowledge Management*, pp. 164–173.

Wu, Y., Wei, F., Liu, S., Au, N., Cui, W., Zhou, H. and Qu, H. 2010, 'OpinionSeer: interactive visualization of hotel customer feedback', *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1109–1118.

Yager, R. R. 1981, 'A new methodology for ordinal multiple aspect decision based on fuzzy sets', *Decision Sciences*, vol. 12, pp. 589–600.

Yager, R. R. 1993, ‘Families of OWA operators’, *Fuzzy Sets and Systems*, vol. 59, pp. 125–148.

Yager, R. R. 1994, ‘On weighted median aggregation’, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 2, pp. 101–113.

Yager, R. R. 2004, ‘OWA aggregation over a continuous interval argument with applications to decision making’, *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, vol. 34, no. 5, pp. 1952–1963.

Yager, R. R. 2005, ‘Some considerations in multi-source data fusion’, in D. Ruan (ed), *Studies in Computational Intelligence (SCI)*, vol. 5, Springer, pp. 3–13.

Yager, R. R. and Rybalov, A. 1996, ‘Uninorm aggregation operators’, *Fuzzy Sets and Systems*, vol. 80, no. 1, pp. 111–120.

Yang, S. J. H., Tsai, J. J. P. and Chen, C.-C. 2003, ‘Fuzzy rule base systems verification using high-level Petri Nets’, *IEEE Transactions on Knowledge and Data Englineering*, vol. 15, no. 2, pp. 457–473.

Ying, M. 2002, ‘A formal model of computing with words’, *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 5, pp. 640–652.

Yoon, K. P. 1996, ‘A probabilistic approach to rank complex fuzzy numbers’, *Fuzzy Sets and Systems*, vol. 80, pp. 167–176.

Yu, J., Amores, J., Sebe, N., Radeva, P. and Tian, Q. 2008, ‘Distance learning for similarity estimation’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 451–462.

Yu, L. and Lai, K. K. 2011, ‘A distance-based group decision-making methodology for multi-person multi-criteria emergency decision support’, *Decision Support Systems*, vol. 51, pp. 307–315.

Zadeh, L. A. 1965, ‘Fuzzy sets’, *Information Control*, vol. 8, pp. 338–353.

Zadeh, L. A. 1975a, ‘The concept of a linguistic variable and its application to approximate reasoning. Part I', *Information Sciences*, vol. 8, no. 3, pp. 199–249.

Zadeh, L. A. 1975b, 'The concept of a linguistic variable and its application to approximate reasoning. Part II', *Information Sciences*, vol. 8, no. 4, pp. 301–357.

Zadeh, L. A. 1976, 'The concept of a linguistic variable and its application to approximate reasoning. Part III', *Information Sciences*, vol. 9, no. 1, pp. 43–80.

Zadeh, L. A. 1996, 'Fuzzy logic = computing with words', *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 2, pp. 103–111.

Zhang, D. and Luqi 1999, 'Approximate declarative semantics for rule base anomalies', *Knowledge-Based Systems*, vol. 12, pp. 341–353.

Zhang, G. 1991, 'Fuzzy continuous function and its properties', *Fuzzy Sets and Systems*, vol. 43, no. 2, pp. 159–175.

Zhang, G. 1992, 'Fuzzy limit theory of fuzzy complex numbers', *Fuzzy Sets and Systems*, vol. 46, pp. 227–235.

Zhang, G. 1994, *Fuzzy Measurement Theory*, Guizhou Sciences and Technology Press.

Zhang, G., Dillon, T. S., Cai, K., Ma, J. and Lu, J. 2009, 'Operation properties and δ-equalities of complex fuzzy sets', *International Journal of Approximate Reasoning*, vol. 50, no. 8, pp. 1227–1249.

Zhang, G. and Lu, J. 2009, 'A linguistic intelligent user guide for method selection in multi-objective decision support systems', *Information Sciences*, vol. 179, no. 14, pp. 2299–2308.

Zhang, Q., Chen, J. C. and Chong, P. P. 2004, 'Decision consolidation: criteria weight determination using multiple preference formats', *Decision Support Systems*, vol. 38, pp. 247–258.

Ziegler, P. and Dittrich, K. R. 2004, 'Three decades of data integration – All problems solved?', in *Proceedings of 18th World Computer Conference*, Toulouse, France, pp. 3–12.