Learning Based Multi-agent Conceptual Ship Design Decision Support System

HAO CUI

University of Glasgow and Strathclyde
Department of Naval Architecture and Marine Engineering

Learning Based Multi-agent Conceptual Ship Design Decision Support System

By

Hao Cui

A thesis presented in fulfilment of the requirements for the degree of Doctor of Philosophy

2010

Acknowledgements

I owe my deepest gratitude to my advisor, Dr. Osman Turan. He provided me support, encouragement and guidance during my PhD study. His vision and insightful thoughts have always thrown me light when I went into dark end. He is more than an excellent advisor, but a mentor and role model.

My special thanks go to Dr. Aykut Olcer. He gave me tremendous help during my study. He helped me to set up the problem in Chapter 8 and provided me guidance and suggestions. Without him, the work in Chapter 8 could not be possible.

I would like to acknowledge great friendship and help from my colleagues: Hassan Khalid, Sulaiman B El-Ladan, Iraklis Lazakis, Emek Kurt, Serkan Turkmen and Tineke Bosma. I also miss the time of department football game every week. My gratitude goes to my teammates: Prof Peilin Zhou, Dr Mahdi Khorasanchi, Giuseppe Mortola and Nabile Hifi. You made my life in Glasgow colourful.

Besides the people who work with me, I want to thank all Chinese friends in the department.

The research in this study is sponsored by British Overseas Research Students Awards Scheme (ORS) scholarship and university scholarship of Strathclyde.

Contents

Nomenclature

CAD	Computer Aided Design
CAE	Computer Aided Engineering
CBR	Case Based Reasoning
EA	Evolutionary Algorithms
GA	Genetic Algorithms
Hs	Significant Wave Height
JADE	JAVA Agent Development Environment
KG	Vertical Centre of Gravity
MAS	Multi-agent Systems
MOEA	Multi-objective Evolutionary Algorithms
NAPA	Naval Architecture Package
NSGAII	A Fast and Elitist Non-dominated Sorting Genetic Algorithms
ROPAX	Ro-Ro Passenger
SEM	Static Equivalent Method
TOPSIS	Technique for Order Preference by Similarity to Ideal Solution
XML	Extensible Markup Language

List of Tables

List of Figures

Abstract

In traditional ship design process, the design work has to depend on the experience of designers. However due to the decreasing number of available experts for various reasons, retaining and the utilising the previous experience/expertise in the industry, which can be design office, shipyard etc, is a serious issue. At the same time, modern ships have become large and complex while trying to comply with stricter requirements of the rules and regulations. The old design support system can not satisfy the design tasks. This research presents a new ship design decision support system with learning ability, which can automatically improve itself with learning. This system can process complex ship design via effective learning by utilising data mining and the available data from natural ship database. It can provide a robust support via automatically learning approaches to help designers to manage their complex world of multiple simultaneous tasks to make an excellent decision and develop innovative design.

In this study, several machine learning methods are applied in different subjects. The decision tree and case-based reasoning is employed to build learning based ship design learning library. The Q-learning method is selected to improve the real-time learning in ship design. The support vector machine based fuzzy multiple attribute decision making method is developed to assist the designers to select the final design. In order to achieve this distributed support system, multi-agent architecture is employed and a new optimization method is created. Two classic problems including stability based hull subdivision design and structure optimization in ship design are studied as case studies. The application of this new methodology demonstrates that this system has excellent performance on both numerical factions and real world problems. The quality of design has been improved greatly with the distinctly reduced time under the assistance of this system.

Chapter 1

Introduction

1.1 Introduction

The global maritime trade is one of the most important activities of the world economics. "*Seaborne trade is, in a sense, at the apex of world economic activity*"(Stopford 1997). The development of maritime trade keeps a large proportion of the world economic and directly affects the growth of the world economic. "*With over 80 per cent of world merchandise trade by volume being carried by sea, maritime transport remains the backbone supporting international trade and globalization.*" (The UNCTAD secretariat 2008). The maritime marketing improves the living of people, enhances the international cooperation and intercommunion, and promotes the development of world economic. The global maritime trade has made a significant contribution to the growth of world economics.

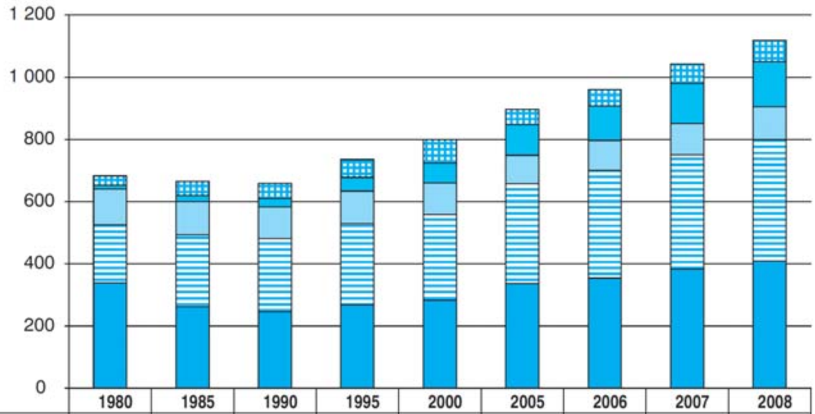
The world's ship fleets provide strong support for the global maritime transport. *At the beginning of 2008, the world merchant fleet reached 1.12 billion deadweight tons (dwt)* (The UNCTAD secretariat 2008). In recent years, the world ship fleet developed rapidly (Figure 1.1) and the demands of fleet expansion require better, faster and cheaper ships.

The increased globalisation brings high level of competition in terms of trade and this reflects on the design and manufacturing of the ships and other floating structures. The ship owners are looking for reducing their cost via cheaper building and operational costs while complying with international standards. However, in recent years, in parallel to other industries maritime industries need to comply with new standards in terms of safety of passengers, crew, cargo, ships and environment. Furthermore, with decreasing fossil based energy sources, new efficient and innovative developments in terms of technologies, designs and energy sources. This means novel ship designs for which designers may not have as much experience as desired.

The new ships should be designed and built to high standards as far as possible to respond to the globe maritime market needs. In the meantime, increasing safety standards with constantly improved ship design and building standards while reducing the cost have been a severe challenge to the naval architects. In order to achieve high performing ship design, a high quality learning based ship design decision support system to assist the ship design work is the key to such achievement. Such decision support system can greatly improve the ability of designers and reduce the design time. Ultimately it can help the naval architects to design the novel and harmonious ship with excellent performance and cost effectiveness.

World fleet by principal vessel types, selected years[a]

(Beginning of year figures, millions of dwt)



	1980	1985	1990	1995	2000	2005	2006	2007	2008
Other	31	45	49	58	75	49	53	63	69
Container	11	20	26	44	64	98	111	128	145
General cargo	116	106	103	104	101	92	96	101	105
Dry bulk	186	232	235	262	276	321	346	368	391
Oil Tanker	339	261	246	268	282	336	354	383	408

Source: Compiled by the UNCTAD secretariat on the basis of data supplied by Lloyd's Register – Fairplay.

[a] Cargo carrying vessels of 100 GT and above.

Figure 1. 1 The DWT of world fleet, taken from the UNCTAD secretariat 2008

(The UNCTAD secretariat 2008)

1.2 Motivation

In current ship design, the existing design system can not store the prior experience of humans and largely depend on software designer/expert interaction. Without the help of prior experience, ship design system can not make an auto-reaction to alteration of design environments but waiting for designer to adjust. It is a hard and time-consuming work which is one of main bottlenecks in ship design.

With the development of ship science and maritime industry, ships have been becoming larger and more complex in order to meet the demands of maritime industry within increasingly global business, leisure and trade. On the other end despite the increasing size of the fleet and shipbuilding, availability of number of experienced/skilled designers is significantly less than the desirable level. As the experience and knowledge intensive subject, ship design largely depends on experience/skills of designers and specialists. The design solutions would alternate according to experience level and therefore the utilization of knowledge/experience becomes new but critically important for modern ship design. Firstly, ship design, as one classic type of systems engineering and parallel design, has to consider the interaction among different design tasks via update of knowledge and experience. Secondly, fast design works up with requirement of maritime market. When enquires or orders come, shipping company not only needs the required knowledge but also fast design to respond to the market requirements, which mean that it is extremely important to design an appropriate ship in a short time. The key factor in fast design is controlling whole process for better design decision making in a most effective way. For this aim, a design decision system must have the learning function to transfer experience to particular ship design to help for design process. Furthermore, the adaptability of dynamic design environment also requires that the design decision system can learn automatically from previous design exercises and provide a real-time answer for questions in new designs. In another word, a design support system with learning function is very useful and essential approach for ship design.

1.3 Problem Definition

As a dynamic process, ship design contains many uncertain information. For decision based design, decision support systems are very important for a successful ship design. The decision support systems should have learning ability, which means the systems can fully use prior experience as new design guidance. However, current decision support systems focus mainly on numerical simulation and calculation, leaving the design experience to the skills of the designers or the specialists who also should be the user of the system. This raises the issue that design quality may heavily depend on a designer’s experience. In order to improve the design quality, as a complex system, new computer aided ship design needs prior experience to avoid failure and to give a better direction. How to make use of this prior experience has become a key problem for the marine design process.

Ship design decision support systems with learning ability will be an important improvement for the current ship design practice. This research aims to improve the ability to process complex ship design via effective learning by utilising the data mining of the ship database together with effective artificial intelligence and practical experience. It can provide a robust support via automatic learning to help designers to manage their complex worlds of multiple simultaneous tasks to make an excellent decision and create an innovative design.

This thesis deal with the development of the learning based ship design decision support system combining optimisation, design and learning approach in an integrated environment. The learning method, artificial intelligence, decision theory, and virtual design are studied and a new approach is developed with a number of applications on various case studies.

1.4 Aim and Objectives of the thesis

Definition of the problem in section 1.3 gives the detailed description of the research problem which needs to be solved in this thesis. This is the preliminary step in research. However, the ship design is a very large and complex subject which requires different types of information and knowledge in both science and engineering areas in an integrated form. Developing a successful ship design decision support system with learning ability is a very challenging indeed. Meanwhile, there have been many other good attempts in ship expert systems and design methods. Inevitably, more effort is required to develop a new system based on various previous research findings. According to this aspect, one of the critical gaps in ship design support system development is the lack of a systemic experience sharing framework to enable self-learning function within the system to solve the practical design problem.

The aim of this research work is to develop a multi-agent ship design decision support system with learning ability, which can automatically improve the design according to the experience gained via agents' self-determination for learning. Through the learning function of agent, the system will abandon line-design frame and transfer to system engineering network which fits better to the dynamic ship design environment.

Therefore the research outcome ought to satisfy the following objectives:

- ✓ Development of learning based ship design decision support framework
- ✓ Development of an integrated learning method, which is suitable ship design problems, in order to find relationship between input and output in ship design process with minimum information.
- ✓ Development of the approach for storing, drawing and using of experience for better management of prior experience.
- ✓ Incorporation of reinforced learning into multi-objective optimisation to improve search performance and better solution space forecasting ability.
- ✓ Improving the speed of optimisation via providing guidance of search direction using learning method.
- ✓ Development of self-learning decision support system to assist the final decision of solutions.
- ✓ Testing and evaluating the framework on existing ship designs and new designs.

In this study, the research will focus on the part of learning ability including experience sharing and real-time learning. The system will accept multi-agent framework, which is studied in previous research. The development of multi-agent framework mainly concentrates on creating a communication environment. For learning method, many kinds of machine learning approaches are introduced and analyzed from both theory and practice in engineering. An integrated systemic learning method is proposed for ship design decision support system.

1.5 Structure of the thesis

The thesis consists of 9 main chapters and a number of associated appendices. The content of each chapter is given briefly as follows:

Chapter 1: Introduction

In chapter 1, a brief introduction is given pertaining to general scope of this research. The critical problem in ship design and the reason for study this problem is explained together with the importance of such things. A brief historical background is introduced while the overall aim of the thesis, structural outline of the thesis is presented. This chapter also presents the definition and explanation of the main aim and objectives of the study.

Chapter 2: Critical Review

In chapter 2, a critical review of ship design, the application of artificial intelligence and machine learning in ship design, ship design optimisation, multi-agent design decision support system and design decision technology is presented. The review focuses on both theory and application together with detail analyses and discussion on the methods.

Chapter 3: Learning Based Ship Design Decision Support System

This chapter presents the blueprint of the whole study covering the global viewpoint from theory to practical application with the functional description. The framework of learning based ship design decision support system is introduced together with the research emphasises and contributions.

Chapter 4: Data Mining in Ship Design

In this chapter, the importance of learning library within the design support system is explained. The reasons of selecting different data mining approaches are discussed. The theory, principle and application of two approaches (decision tree and Case Based Reasoning (CBR)) are given. This chapter provides the method of drawing the experience, which includes the selection of design variables, the choosing of optimisation objectives and the range of constraints etc., from previous cases to better improve the next design. The research focuses on the method to build the new database, named learning library which uses the relationship storing to replace traditional data storing. The decision tree and CBR are employed in this chapter to deal respectively with the numerical and linguistic attributes.

Chapter 5: Multi-objective Optimisation and Multi-objective Particle Swarm Optimisation in Ship Design

Optimisation is the core part of decision based ship design theory. The optimisation can provide strong support to decision maker. In this chapter, a new optimisation approach, multi-objective particle swarm optimisation based on multi-agent system, is introduced. The comparison of the proposed method with NSGAII algorithm is carried out to evaluate this new approach on the test functions and real cases. The aim of this chapter is to provide an easy parameter setting optimisation algorithm with good inside learning ability.

Chapter 6: Real-time Learning in Ship Design Environment

Time cost is a critical problem for optimisation part in ship design support system. Real-time learning, which can change optimisations own behaviour according to the practical environment, is proposed in this chapter to solve time cost problem and help proposed system to obtain better solutions. The real-time learning can forecast the optimisation direction based on prior experience in the running. Meanwhile, the real-time learning can prevent the optimisation into inefficient areas. This chapter introduces the Q learning, which was proposed first time by Watkins (Watkins 1989), as one of the real-time learning approaches, then explains the algorithm and presents an application using barge model.

Chapter 7: Learning Based Ship Design Decision Making and Decision Support System.

In this chapter, the fuzzy multiple attribute decision-making (FMADM) is employed to solve the problem of selecting final decision. The original FMADM is rebuilt via multi-agent system to automate for improving the adaptability to large scale problem. But when FMADM is applied on ship design practice, the main problem, which is difficult to be solved in the traditional method, is that the human specialists are very hard to be selected and ranked. In this chapter, a new Learning and Multi-agent based FMADM (LMFMADM), which adds the learning function to assist FMADM to adjust the situation of where there is a lack of enough human specialists, is proposed. A new machine learning approach, support vector machine, is employed to realize the self-learning of system in this part and assign virtual specialist committee and technology manager, who will rank the specialists. The procedure of proposed learning based ship design decision system is concluded and analysed. The detailed description about structure and function of every part is presented.

Chapter 8: Case Study

In this chapter, two cases are employed to evaluate proposed system. The first case study is a classic stability design problem. The different algorithms are utilised to compare and the learning function is added. In the second case study, a chemical tanker structure design problem based on CBR rules of IACS is selected to evaluate the proposed system.

Chapter 9: Discussions and Conclusion

This chapter reviews the whole research including the account of the original contributions and achievements of this thesis. The discussions are outlined while further considerations are given for possible future work on the basis of experience gained during this study. This chapter contains main conclusions emerging from this research study.

Chapter 2

Critical Review

2.1 Introduction

The learning based design decision support system is a multidisciplinary approach, which can improve the practical ship design significantly. It develops an integrated system by combining engineering design, human decision factor, computer science etc. In this research, ship design methods, decision making approach, decision support system, artificial intelligence and machine learning, multi-agent system and multi-objective optimisation are studied and therefore in this chapter, the developments with regard to these subjects in ship design area are critically reviewed. As many of these subjects have very rare application in ship design, the review will make reference to related engineering applications in order to give a full picture of these technologies used in this thesis.

This review mainly focuses on the new ideas developed in recent years. The multi-agent system puts emphasis on the system framework and conflict resolution. Because there is little application of machine learning in ship design, the review is extended to the intelligent systems in ship design, which utilise some machine learning approaches. The optimisation in ship design is a traditional research area and attracts plenty of research interests. The review of this part focuses on the multi-objective optimisation via GAs and new Particle Swarm Optimisation (PSO) algorithm in practical application.

2.2 Ship design method

The traditional ship design method, which is still broadly practised in most ship design departments and shipyards, will be mentioned but the more emphasis will be laid on the new advanced ship design methods. The detailed analysis will be given to provide an insight of the features of new system.

There are many different stages in ship design which may be called under different names due to the different design phases such as navy ship and merchant ship. At the same time the key points of design are different between navy ships and merchant ships. This is also one of the reasons why there are different definitions of design stages. The design stages in this thesis are categorised as shown in Figure 2.1. The Basic Design includes Concept Design, Preliminary Design and Initial Design (Contract Design and Functional Design).

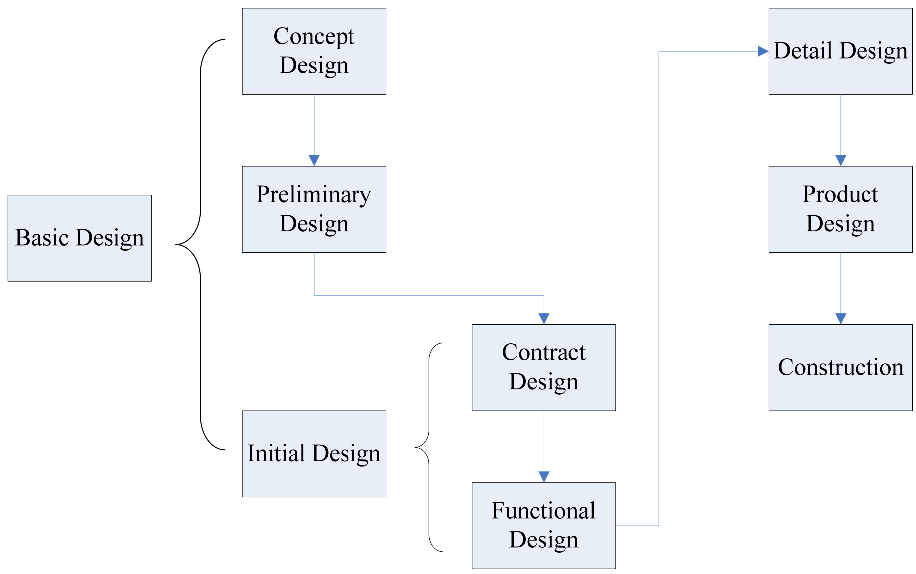


Figure2. 1 Ship Design Phases

2.2.1 Traditional ship design methods

The "modern ship era" began in the mid 1800's. Before that, the ships were usually designed by simply developing a new ship based on an existing one. The new ship would be the scale of old designs with minimal alterations, so the results would be random because no scientific principles were applied.

Normally, it is considered that the modern ship era begins with two significant events: the replacement of sail power with steam powered propulsion in 1780s and William Froude's scientific approach to vessel performance prediction in 1860s.

The design spiral, proposed by Professor J. Harvey Evans (Evans 1959) has been used to describe the preferred ship design process for many years as shown in Figure 2.2. In this method, ship design is viewed as a sequential iterative process (design spiral) and in every phase, ranging from concept design to detail design, every important aspect of the ship design is re-evaluated, starting from mission requirements to hydrostatics, powering and cost estimates. In every cycle of the design spiral, the complexity increases, however, the number of possible designs decreases.

Buxton (Buxton 1972) proposed to embed cost estimates into the design spiral, particularly at the preliminary design stage. Buxton pointed out that the benefits of including cost estimates into the design spiral are easy comparison among different concept alternatives, and to reduction of the time required for tendering.

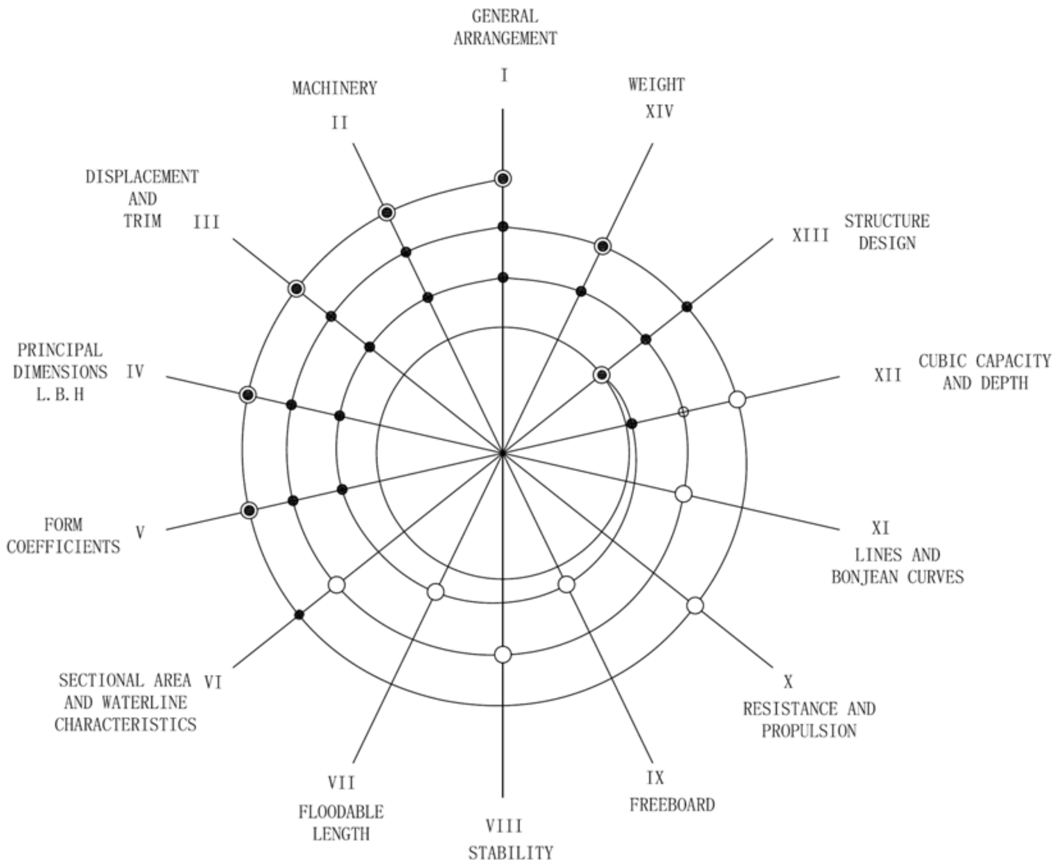


Figure2. 2 The Ship Design Spiral, from (Evans 1959)

Andrews (Andrews 1981; Andrews 1998) proposed to include constraints into ship design spiral in three categories: design, design process and design environment. The minimizing building time, structure of design organization and economic climate can be listed as constrain respectively for each category. Andrews also discussed the layout arrangement of a ship and size consideration of the ships according to volumetric considerations, major spaces, machinery spaces, and residues. In a later study, Andrews proposed a more sophisticated approach to ship design, by including mission information, as shown in Figure 2.3. The methodology was called building block methodology and the functionality of each component was defined in the design process while the methodology was realized by commercially available ship design software.

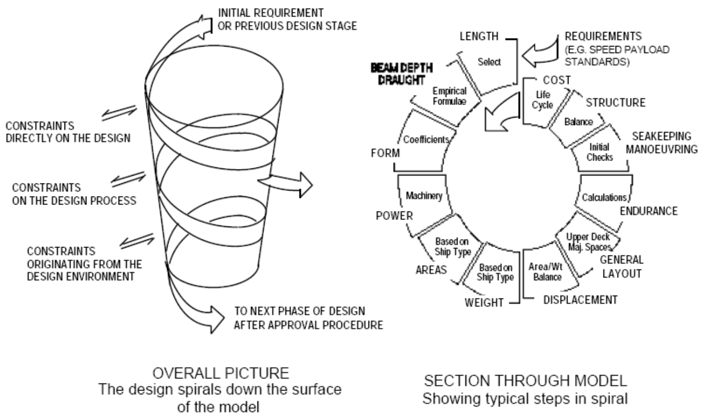


Figure2. 3 Overall module of ship design process (Andrews 1998)

2.2.2 New ship design approaches

2.2.2.1 Decision based ship design

Mistree et al. (Mistree, Smith et al. 1990) proposed a contemporary paradigm, decision-based design, for the design of ships which encompasses systems thinking and embodies the concept of concurrent engineering design for the life cycle. The author analyzed the traditional ship design spiral and pointed out its disadvantages. The author considered that the spiral was effective only when the shipbuilding industry was doing well and the volume of ship building was high, but the spiral was ineffective when the market was depressed and single ship design was done. The author also criticized that the spiral was limited to single objective optimisation. In order to overcome these disadvantages and to enable concurrent engineering practice, Mistree proposed a frustum of cone model instead of a design spiral, as shown in left part of Figure 2.4. In the outer surface of the cone, the spiral proceeds along with constraints. For the cross section, Mistree adopted a concurrent engineering approach, which is represented by rings of interaction depicting information flow to different aspects of ship design such as structure, stability and lines. The right part of Figure 2.4 showed the systems approach used in the paper. Mistree also proposed design guidance system to implement decision-based design and argued to implement design changes by resolving decision problems in phases of the design process. A light-patrol frigate was used as a case to evaluate this decision based design method.

Mistree et al. introduced the decision based design into ship design area and realised that the concurrent engineering was the trend of ship design. The author also forecasted that the role of computer would change from the tools to the partner of design work, and put the designer on the role of decision maker. However because engineering theory and computer technology on ship design area of twenty years ago were not well developed, the author did not recognize that the multi-agent system was a strong support tool for decision based design theory but tried to use simple computer technology to support the complex design situation. This increased the difficulty of realizing this method. At the same time, the author also ignored the importance of experience. The experience of the designers determines the quality of design decision. Without experience, it is very difficult for automatically updating system to provide the full support for decision maker.

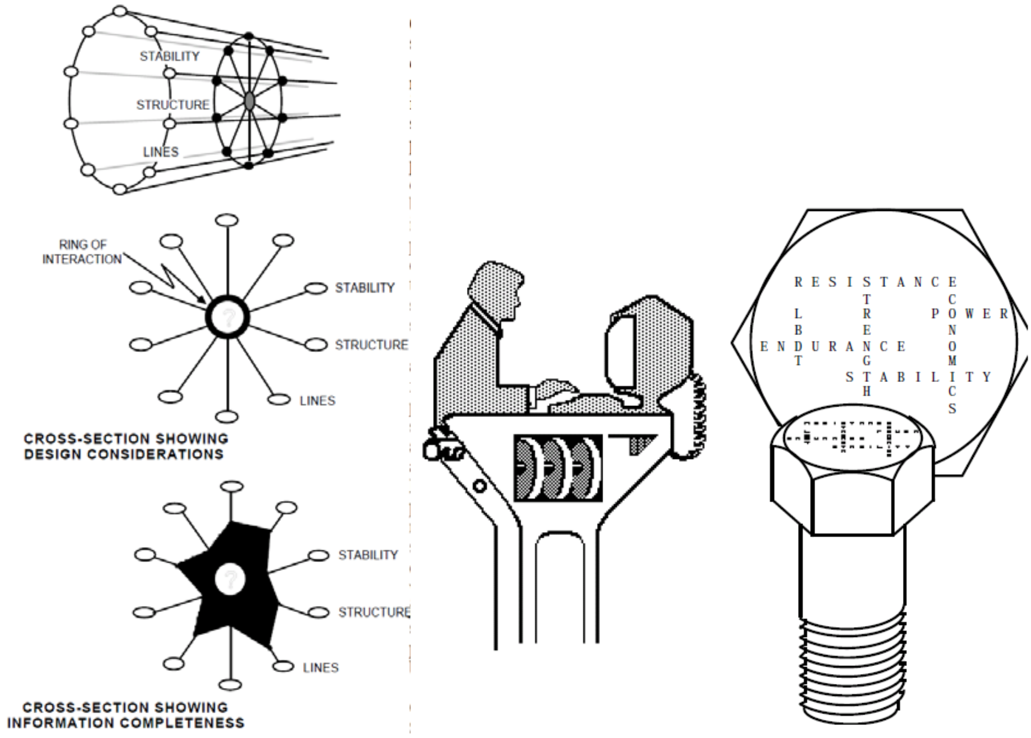


Figure2. 4 Decision based ship design approach (Mistree, Smith et al. 1990)

2.2.2.2 Set-based ship design

Parsons et al (Parsons, Singer et al. 1999) introduced the set-based design to ship design. They proposed a hybrid human computer agent approach to facilitate set-based conceptual ship design by a cross functional team of naval architects and marine engineers. The authors found that the advanced marine design, particularly in the United States, advocates the use of cross-functional design teams, or Integrated Product Teams (IPT's), who will undertake a concurrent engineering approach to all phases of ship design. Further, the study of the world-class Toyota automotive design process had highlighted the potential of a set-based design approach in concurrent engineering to provide a greater probability of achieving a global optimum for the overall design. The disciplinary/technical specialists were organized and acted as agents within a design network that can be either co-located or interconnected across the web. Computer agents are introduced between each pair of human design agents to facilitate their communication and negotiation. A systematic market approach, developed in the Defense Advance Research Projects Agency (DARPA) sponsored Responsible Agents for Product-Process Integrated Design (RAPPID) project, was utilized as an initial approach to facilitate this team for set-based design.

The author introduced the agent based approach for realizing the set based ship design and successfully applied it to a real case. But the agent in this paper was hybrid agent, which meant the agent was consisted by computer and human together. This would reduce the robustness of the design system and decrease the speed of whole design process. The second disadvantage was the chief agent in the agent framework. Obviously, the chief agent was a simulation of the general manager in real world, but when all the information was sent to the chief agent, the whole design work had to wait for the chief agent to take action. This might have caused the problems from information sharing to process blocking in the practical design work.

2.2.2.3 Risk-based Ship Design

Applications of risk-based approaches in the maritime industry started in the early 1960s with the introduction of the concept of probabilistic ship's damage stability. (Papanikolaou 2009)

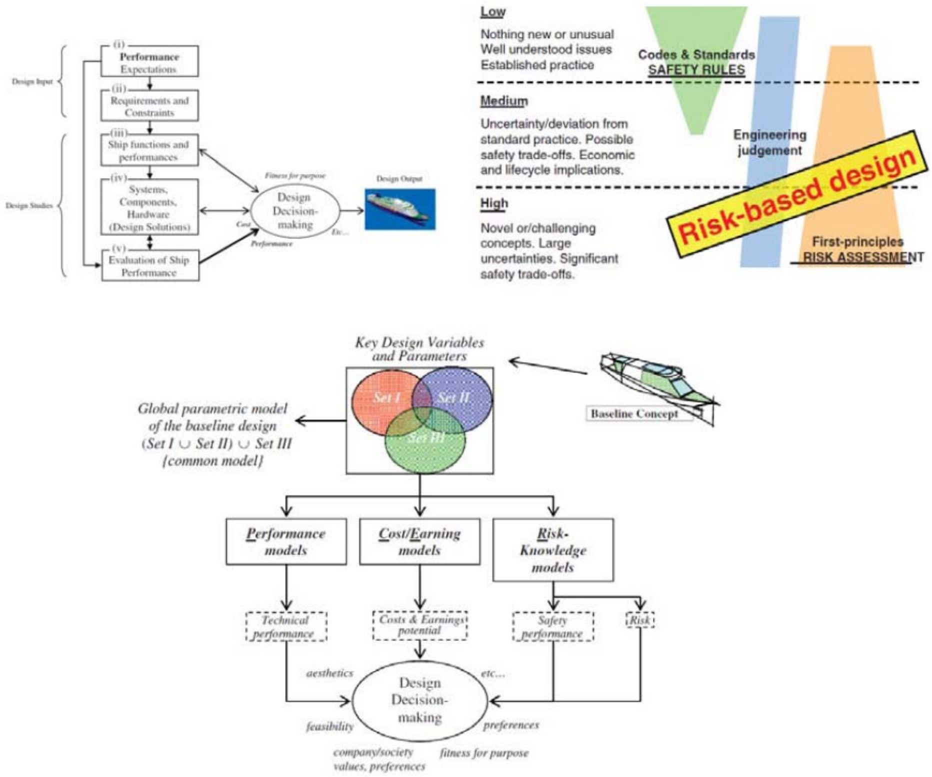


Figure2. 5 Risk based ship design approach (Papanikolaou 2009)

The risk based design is developed from rule based design as shown in Figure 2.5. This method supports a safety culture paradigm in the ship design process by treating safety as a design objective rather than a constraint. It creates a design process for the concept design stage under extremely tight cost and time constraints. The method points out that notion of "risk" is usually associated with undesirable events and shipping operations being undoubtedly "risky". So the ships should be designed with this as basic principle.

This method aimed at fast design via matching the limitation of the rules and all kinds of design conditions. For solving the conflict caused by extremeness of design limitations, this method put the safety at the first place. This method will fit into the economic requirements of the maritime industry but do not provide insight details of the ship design. The harmonious of very design factors are equally important to economic aims. So the multi-objective optimisation and decision making approach should be fully developed for this method.

Table 2.1 Development of ship design methods

Method	Presenter	Years	Main Ideas
Design spiral	Evans, J. H	1959	Ship design is a sequential iterative process; every important aspect of the ship design is re-evaluated to improve the complex and decrease the designs;
Design spiral with cost estimates	Buxton, I.L.	1972	Embed cost estimates into the design spiral;
Design spiral with constraints	Andrews, D	1981	Included constraints into ship design spiral in three categories: design, design process and design environment; the building block methodology is added to design spiral;
Decision based design	Mistree, F.	1990	Introduced the decision based design into ship design area and realised that the concurrent engineering is the current of ship design,designer is decision maker
Set based design	Parsons, M. G.	1999	Introduced the set based ship design and the agent based approach
Risk based design	Vassalos, D. Papanikolaou A.	2009	Aimed at fast design using the limitation of the rules and all kinds of design conditions and put the safety at the first place

Table 2.1 summaries the development history of main ship design methods in recent years.

2.3 Multi-agent system application in ship design

Multi-agent design system (MAS) was proposed in 1990s and has developed very quickly in recent years. There are many kinds of definitions of multi-agent system and the following definitions are used in this thesis. An agent is anything that can perceive its environment through sensors and act upon that environment through actuators (Russell and Norvig 2003). A system that consists of a group of agents that can potentially interact with each other is called a multi-agent system (MAS), and the corresponding subfield of artificial intelligence (Eamon and Rais-Rohani) that deals with principles and design of multi-agent systems is called distributed AI (Vlassis 2007).

The agents in a multi-agent system can be robots, human or human teams, but in this study, they represent the software agents. The multi-agent system can deal with a complex task even when its individual strategies are simple. Based on this characteristic, there are several applications of MAS in ship design.

One of the first ship design studies of MAS is presented by Parsons et al. (Parsons, Singer et al. 1999) as shown in Figure 2.6. The authors built a hybrid network of human and computer agents to solve the complex environment such as ship design which continue to require the expertise, perception and judgment of the human designers. This MAS method is keeping the close connection with the set based design in subsection **2.2.2.2**. A systematic market approach for design negotiation was studied in this paper. The author also gave the definition and function of agents as follows: the agents are elements of computer code with elements of perception, intelligence and adaptability capable of taking independent action. In the paper, there were seven agents being created: chief engineer agent, resistance agent, maneuvering agent, stability agent, cargo agent, propulsion agent and hull agent, as shown in Figure 2.6. The chief agent is set as the head agent, which is the overall leader of design team and serves as the voice of the customer. The author used a market approach to find the design solutions. The agents participate in a design marketplace where the goods being traded represent the design characteristics in a common currency. The conceptual design of a hatch-covered, cellular feeder container ship was undertaken by a team of student design agents to assess the effectiveness. Singer and Parsons (Singer and Parsons 2003) developed their idea and proposed a fuzzy logic software agent model. A new implementation of a fuzzy system has been developed to enable the designers to communicate their preferences over a range of parameter values. The authors added a fuzzy software agent on the model of (Parsons, Singer et al. 1999). In the paper, the chief engineer agent is still the control agent and fuzzy logic software gives the chief engineer suggestions for the possible range of a variable for a new negotiation round, but the ultimate decision is in the hands of the chief engineer.

The author introduced the agent concept to ship design and discussed the negotiation method via a market approach. But the whole process of the design was controlled by the chief engineer agent. This would limit the effect of MAS. Because all the information would be sent to the chief engineer agent, the information can be blocked at this point. The chief agent must have the outstanding ability to take charge of whole design. This still made the design quality to largely dependent on the personal ability.

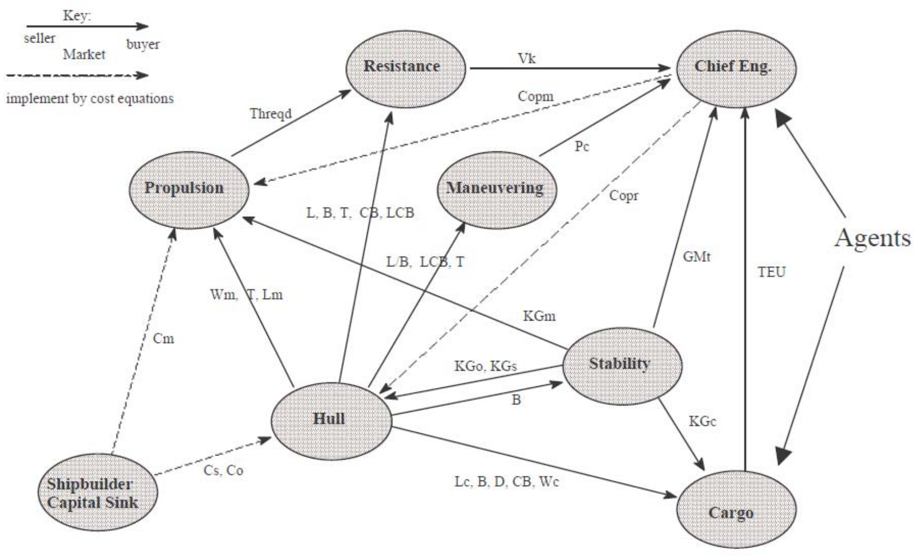


Figure2. 6 The agent system network structure (Singer and Parsons 2003)

Another design system for ship design was proposed by Fujita et al. (Fujita and Akagi 1999) as shown in Figure 2.7. The authors gave the method for agent-based distributed design system and also gave the computer implementation of distributed design system. For application, an experimental system for basic ship design was proposed and a 38,000 ton deadweight bulk carrier was employed to evaluate this system. For agent communication, the authors defined four kinds of messages based on object orientation: send-type message, broadcast-type message, agent-setup-type message and remote-procedure-call-type message. The computer environment was UNIX and programming language was Allegro Common Lisp. For experimental system, the authors mentioned that the design procedures and knowledge had been well established.

In this paper, the author introduced the software agent into ship design and also built the distributed design system. However for the knowledge sharing, the author did not give enough information and did not give the detailed building procedure. The author proposed communication mode but this method was based on single direction specific objective. This meant the agents still were organized via line process and can not adjust to the dynamic environment.

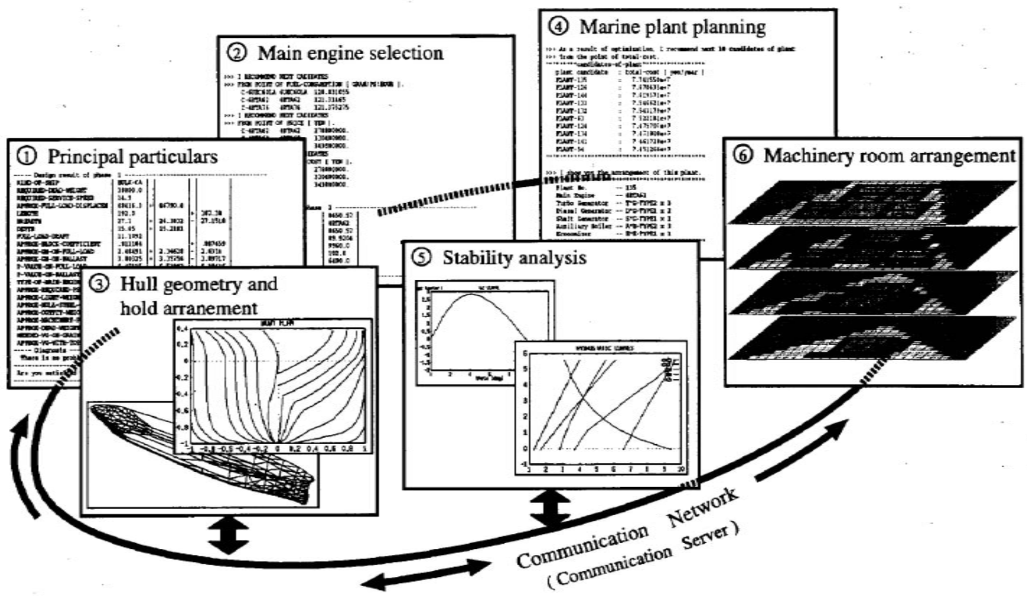


Figure2. 7 The distributed basic design system for ship (Fujita and Akagi 1999)

Lee et al. (Lee and Lee 2002) proposed an agent-based system and focused on the negotiation part among design agents as shown in Figure 2.8. They introduced agent-based collaborative design system and conflict resolution based on a case-based reasoning approach. Under the concept of a global economy, design and production may be assigned separately to different companies. In order to overcome the problem of low productivity due to the interruption of information, the concept of simultaneous engineering and concurrent design becomes very significant. In this article an agent-based ship design system was developed to support cooperation in distributed ship design environments. The authors pointed out that the conflicts that occur while sharing knowledge in the system must be resolved. One approach is to adopt a case-based conflict resolution strategy formulated to resolve current conflict on the basis of similar previous cases in agent-based collaborative design system environments. In order to do this, conflict cases that occur in the initial ship design stage are provided. The case-based conflict resolution strategy was evaluated by applying it to a collaborative design process in the initial ship design stage, especially the machinery outfitting design, the preliminary design, the hull form design, and the structural design. Through the help of the collaboration of the design agents, the facilitator, the conflict resolution handler, and the case-based system, a designer can make decisions based on similar previously resolved cases.

The author recognized the importance of collaboration of the agents and used a case based reasoning method to solve the conflict of the agents. But the author did not provide the insight of the relationships what case-based reasoning was found. Actually, what CBR gives are the experiences of prior work and depending only on the CBR to seek sharing of the experience may greatly reduce the speed of system. The author did not study the running time for large design problem.

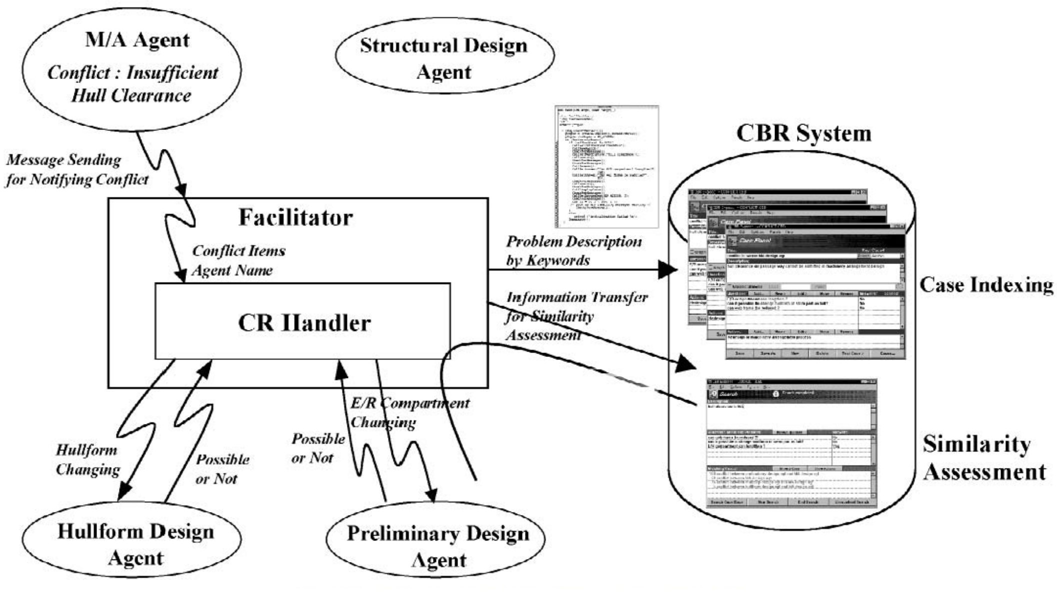


Figure2. 8 The conflict resolution process in the collaborative ship design agent system (Lee and Lee 2002)

Turkmen and Turan (Turkmen and Turan 2003; Turkmen and Turan 2004; Turkmen 2005) developed a new multi-agent ship design decision support system as shown in Figure 2.9. The ship design agent group concept was defined and a case study of hull subdivision problem was given. The authors created five agents in the study: Deck Layout Agent (DLA), Damage Stability Agent(DSA), Economics Agent (Trelea), General Arrangement Agent (GGA) and User Interface Agent (UIA) to solve a damage stability design problem as shown in Figure 2.9. The JADEX agent design environment was employed. A subdivision of barge was employed to evaluate the system. The authors provided the framework of multi-agent system for ship design and the work is the basis of the study presented in the thesis.

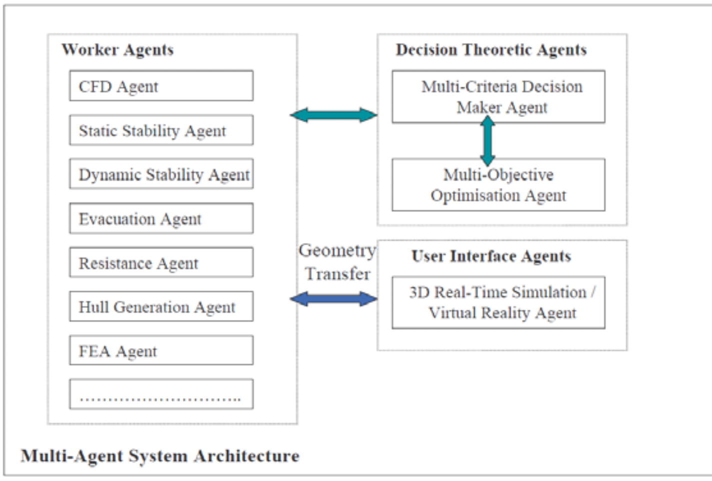


Figure2. 9 The multi-agent architecture (Turkmen 2005)

2.4 Machine learning and intelligent system in ship design

Herbert Simon (Simon 1983) defined machine learning as ‘Any changes in a system that allows it to perform better the second time on repetition of the same task or on another task drawn from the same population’. In this study, machine learning means the design decision method can draw experience from prior designs and give directions to current design for better design quality. The tasks of machine learning in this design decision system focuses on two aspects: one is to draw design rules and limitations from prior designs and the other is to help designers to adopt dynamic environments for optimisation.

Machine learning has been developed rapidly in recent years. But the application of machine learning in ship design is very rare. Although some intelligence systems or design systems utilise one or several approaches of machine learning, the systematic application is very limited in literature.

The review of this part will not be just limited to machine learning but also extended to other expert systems which use one or several learning approaches in ship design to give a review of developments on artificial intelligent application in ship design.

2.4.1 Machine learning of ship design system

Srdoč et al. (Srdoč, Bratko et al. 2007) proposed a machine learning approach in ship repair domain. The authors used machine learning to leave values of the target attribute as they are, and use learning schemes for numerical prediction. The regression and model trees, which is a variant of decision trees was employed and for comparison reasons, instance based learning—another machine learning approach, and classical statistical approaches, had also been used in the study. The software Weka (Witten and Frank 2005) was employed for learning approaches. Weka contains the data mining tools for data pre-processing, classification, regression, clustering, association rules, and visualization.

This work introduced regression and model tree and instance based learning into the ship repair programming. But the application was still in the traditional machine learning area --- programming problem and the learning approach was also relatively simple.

Alkan et al. (Alkan and Gülez 2004) developed a knowledge-based computational design tool for determining preliminary stability particulars of naval ships. A robust neural network (NN) structure was established and using principle design data from 22 naval ships. This NN structure used both the classical back-propagation algorithm (CBA) and the fast back-propagation algorithms (FBA). The ANN method has the problems for small sample and over fitting problem which are not discussed by that the author often happen in ship design area. The author also did not provide the detailed training information.

Lee et al. (Lee, Oh et al. 2006) dealt with generating optimal polynomials using genetic programming (GP) as the module of Data Miner. The Data Miner for the ship design based on polynomial genetic programming was presented. The paper dealt with polynomial GP for regression or approximation problems when the given learning samples were not sufficient. The authors generated 1000 data by using empirical formula and the data contained some noise. The system automatically used 800 data for training, and 200 data for test among 1000 learning data. The system was implemented by using Microsoft Visual Studio .Net C# programming. The author said that because of security problem, the paper could not provide the detailed information. So it is impossible to review the detailed method.

From the review, it can be seen that the machine learning in ship design is at the starting stage. The application depends on the statistics software and just made some very simple applications. The machine learning approaches employed by these applications are very limited.

In the following part, the learning approaches used in this thesis will be reviewed. Because this thesis focuses on learning application in ship design, the review correspondingly put emphasis on learning approach application in ship design area.

2.4.2 Decision tree and its application in engineering

Decision tree is one of the most popular learning approaches. Decision tree is the tool which uses tree-like graph or model to classify instances by sorting them based on feature values. Feature values in decision tree means the values which can represent the characteristics of the case. Each node in a decision tree represents a feature in an instance to be classified, and each branch represents a value that the node can assume.

Instances are classified starting at the root node and sorted based on their feature values.

With the development of the decision tree theory, there are different versions (for example ID3, C4.5, C5 etc.) of algorithm for different stages and also there are different varieties for every version. Tjen-Sien Lim (Lim, Loh et al. 2000) made a comparison between decision tree and other learning algorithms and shows that C4.5 has a very good combination of error correction and speed. Figure 2.10 gives the work flow of C4.5 algorithm.

The pseudo code of C4.5 algorithm

1. *Check for base cases*
2. *For each attribute a*
3. *Find the normalized information gain from splitting on a*
4. *Let a best be the attribute with the highest normalized information gain*
5. *Create a decision node that splits on a best*
6. *Recur on the sublists obtained by splitting on a best and add those nodes as children of node*

Figure2. 10 Work flow of C4.5 algorithm (Quinlan 1993)

There are three main reasons why the decision tree is selected as a learning approach in SDLL. The first one is the decision tree has good ability to operate complex representation and also can be easily explained. In the decision support system, one important principle is that the analysis process should be powerful and easily understood. The designers not only want to know the calculation results when the system solves a complex problem but also the process to make the results. In another word, how to get the results is much more important than the solution itself. Normally，only when the designers clearly understand the mechanism, they will apply it in practice. The decision tree can make the decision makers understand the process of analysis better and can use it in application.

The second one is the ability of treating the discrete data. In most ship design optimisations, the objectives and limitations are discrete. The learning approach needs very good ability of processing discrete data together with the ability to deal with continuous data.

Table 2.2 The comparison of main data mining approaches

	Decision Trees	Neural Networks	Naïve Bayes	kNN	SVM	Rule-learners
Accuracy in general	**	***	*	**	****	**
Speed of learning with respect to number of attributes and the number of instances	***	*	****	****	*	**
Speed of classification	****	****	****	*	****	****
Tolerance to missing values	***	*	****	*	**	**
Tolerance to irrelevant attributes	***	*	**	**	****	**
Tolerance to redundant attributes	**	**	*	**	***	**
Tolerance to highly interdependent attributes (e.g. parity problems)	**	***	*	*	***	**
Dealing with discrete/binary/continuous attributes	****	***(not discrete)	***(not continuous)	***(not directly discrete)	**(not discrete)	***(not directly continuous)
Tolerance to noise	**	**	***	*	**	*
Dealing with danger of overfitting	**	*	***	***	**	**
Attempts for incremental learning	**	***	****	****	**	*
Explanation ability/transparency of knowledge/classifications	****	*	****	**	*	****
Model parameter handling	***	*	****	***	*	***

Table 2.2 compares the performance of some main data mining approaches. In the table, one star means poor and two stars means normal when three stars means good and four stars means excellent. The ability of decision tree to deal with discrete data is outstanding and it can also operate the continuous data, so it is very suitable for requirements of the study in this thesis.

The last one is the speed of classification. The fast speed can reduce the time of the training and running, especially for mass data. In modern marine design, time is very important parameter for generally high quality design. The decision support system requires the learning algorithm to quickly finish classification and give correct classification in time. The speed of classification of decision tree is relatively fast in all the approaches of data mining as shown in Table 2.2. This proves that the decision tree can successfully solve the time costing problem in running process.

As the expression of the decision tree, an effective and intuitive tree is selected and this will help the designers to understand the process of making decision. Each inner node corresponds to a variable; an arc to a child represents a possible value of that variable. A leaf represents the predicted value of target variable given the values of the variables represented by the path from the root.

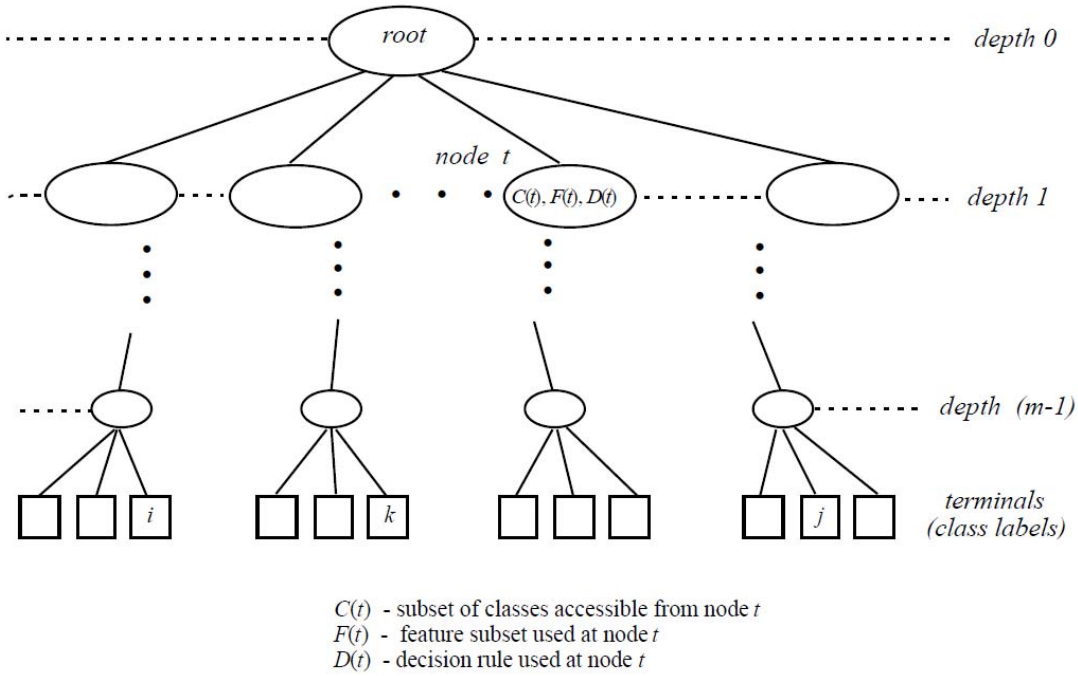
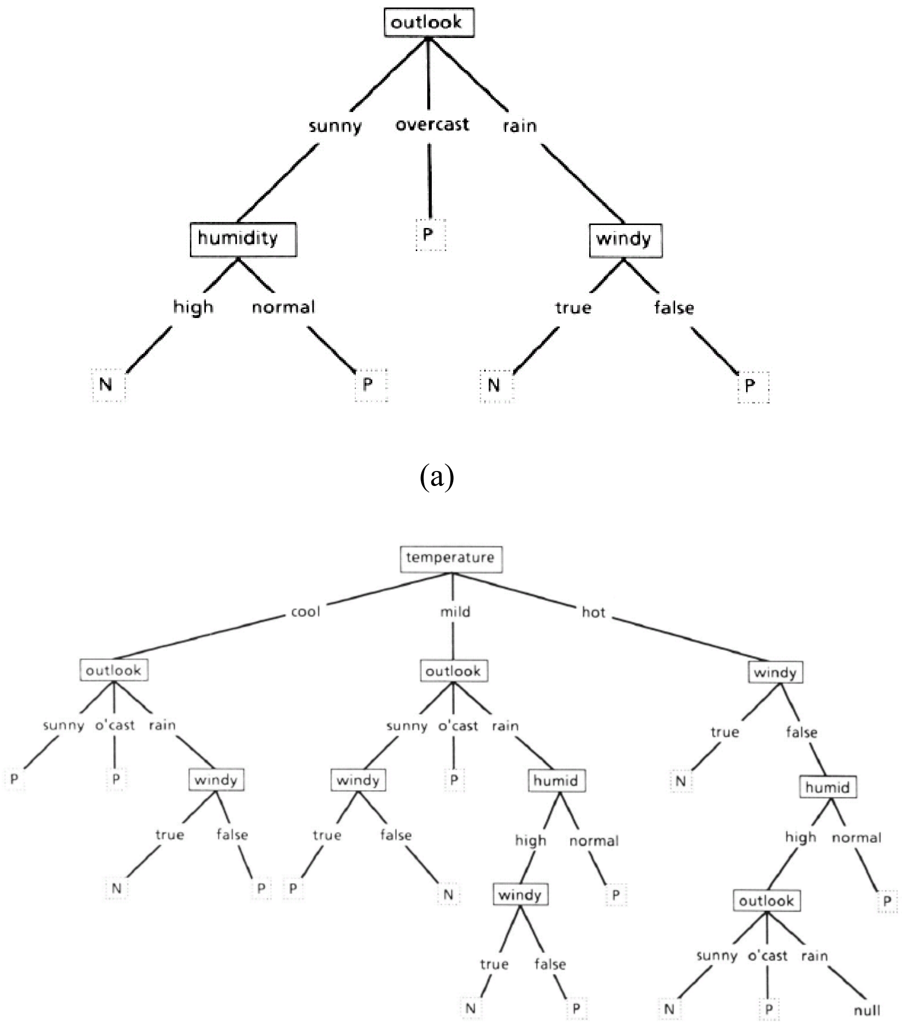


Figure2. 11 Example of a general decision tree (Safavian and Landgrebe 1991)

Figure 2.11 gives an example of a general decision tree including the root, node, leaf (terminals). Figure 2.12 gives more detailed examples of decision tree. Figure 2.12 (a) is an example of simple decision tree and (b) is an example of complex tree.

The study of applying decision tree to ship knowledge is very limited while its application in knowledge based ship design is very rare. Caprace et al. (Caprace, 2007) proposed using decision tree to process the ship data as a tool, which is used in classification problems. The authors point out that" *This method has the major* *advantage to select the most relevant variables before an analysis by an artificial neuronal network in order to avoid unnecessary high computing times*"



(a)

(b)

Figure2. 12 Examples of decision trees(Quinlan 1986)

2.4.3 Case-based reasoning and application in engineering

The case-based reasoning (CBR), also called as instance learning, developed since 1977, is one of the important learning approaches. The idea of CBR is simple and useful. It uses the solutions of past problems to derive the solutions for new problem. In concept, CBR is to solve a new problem by remembering a previous similar situation and by reusing information and knowledge of that situation. (Aamodt 1994)

With many years of development, the case-based reasoning approach has formed a standard four-step process.

Retrieve: Given a target problem, retrieve cases that are relevant to it from memory. A case consists of a problem, its solution, and, typically, annotations about how the solution was derived. For example we want to design a new tanker with length of 150m, we can look for in the database that successful designs with length from 120 m to 180 m and find the nearest design as the reference one.

Reuse: Map the solution from the previous case for the target problem. This may involve adapting the solution as needed to fit the new situation.

Revise: Having mapped the previous solution for the target situation, test the new solution in the real world (or a simulation) and, if necessary, revise.

Retain: After the solution has been successfully adapted to the target problem, store the resulting experience as a new case in the memory.

Lee (Lee and Lee 2002) used case-based reasoning to solve the conflict of agent-based system, which was reviewed in part 2.3.

Delatte and Butler (Delatte and Butler 2003) proposed an object-oriented model for conceptual ship design supporting case-based design. The authors presented a data storage system to store historical design data for subsequent reuse in conceptual design. The database was designed to support case-based reasoning and other similar processes in which recall of past solutions becomes a basis for adaptation to form a new solution. The data involved complex geometric information, and an object-oriented database system was presented. The authors gave detailed application of the design information but the revision of 3D model needs further study. The 3D model of CBR in this article should be developed for practical application.

Kowalski (Kowalskia, Meler-Kapciab et al. 2005) introduced CBR methodology application to aid ship's engine room automation design. The paper presented case based reasoning (CBR) methods of cases in which similarity calculation is applied in the elaborated expert system for the design of ship's engine room automation. The author proposed that CAD model can be used but did not give the model examples.

2.5 Multi-objective Optimisation in Ship Design

Multi-objective optimisation is one of the main study areas of ship design. Especially in recent years, there are many multi-objective optimisation approaches being used in ship design area. In this study, a new multi-objective particle swarm optimisation approach is proposed. Some previous work of the PSO algorithm and multi-objective optimisation used in ship design are reviewed in this section.

2.5.1 Multi-objective genetic algorithm

MOGA (multi-objective genetic algorithm), as one kind of heuristics methods, looks like the only solution to the complex ship design optimisation problem. (Sen and Yang 1998). Because there are many researches of MOGAs in every area of ship science, this review mainly focuses on the application of main MOGA methods on the ship design.

Lee (Lee 1997) proposed a hybrid optimizer for marine vehicle design with aid of design knowledge. The hybrid optimizer is constituted by the genetic algorithm and direct search method. Then a knowledge processor gives the knowledge support. The author recognized that the main weakness of the genetic algorithm is that it requires longer time because of the search of many points. The author wanted to find out a candidate region around the global optimum point by using the genetic algorithm, and then searching the global optimum point in the region by direct search. The task of the knowledge-based system in this paper is to improve the efficiency of optimisation by the generation of proper input data for the design model before performing optimum design process. The author mentioned that knowledge based system provides population size, number of generations, crossover and mutation rate for the genetic algorithm and proper step size and end conditions for direct search method. But the author did not give how the knowledge based system works and gave references in Korean, so it is very difficult to define the real benefits of proposed approach.

Finding some possible optimal points and then performing a small search for finding final solutions can be seen as one kind of disguised ε-constraint methods.

It is noteworthy that this paper recognized the importance of small population and generations for ship design process. The author tried to find optimal candidate solutions to reduce the calculation time. But in multi-objective optimisation, it is very difficult to find these candidate solutions and if the candidates are the wrong choices, the whole optimisation will fail. On the other hand, this method can find several different local optimal points and the final solution is on or besides one of these points. This is also difficult for multi-objective optimisation technology.

Thomas (Thomas 1998) used Pareto ranking, MOGA and NPGA, to investigate the feasibility of full stern submarines. Three objectives were considered: maximize internal volume, minimize power coefficient for ducted propulsion submarines, and minimize cavitation index. Binary representation and different selection techniques were used. Thomas also compared several different algorithms and reached conclusions that MOGA outperforms the other methods in all of the aspects considered.

Pareto efficiency, or Pareto optimality, is an important concept in economics which was used in multi-objective optimisation. The term is named after Vilfredo Pareto, an Italian economist. Informally, Pareto efficient situations are those in which any (additional) change to make any person better off is impossible without making someone else worse off. In the multi-objective optimisation, the designers are looking for solutions for which each objective has been optimized to the extent that if try to optimize it any further, then the other objective will suffer as a result. These solutions are called Pareto solutions.

Brown and Thomas (Brown and Thomas 1998) used a GA with Pareto ranking for naval ship concept design. Two objectives are considered: maximize overall measure of effectiveness (this factor represents customer requirements and relates ship measures of performance to mission effectiveness) and minimize life cycle cost. Binary representation and roulette wheel selection with stochastic universal sampling were used. Brown et al. (Brown and Salcedo 2003; Brown and Mierzwicki 2004) introduced a multi-objective genetic optimisation in naval ship design. A Multiple-Objective Genetic Optimisation (MOGO) is used to search design parameter space and identify non-dominated design concepts based on life cycle cost and mission effectiveness. A non-dominated frontier and selected generations of feasible designs are used to present results to the customer for selection of preferred alternatives. The genetic or evolutionary algorithm used decimal floating-point gene coding and a finite resolution and range or set of values for design variables. A national guided missile destroyer (DDGx) is selected as a case study and overall mission effectiveness (OMOE) and life cycle cost (LCC) are moulded as the objective attributes. The optimisation was run for 100 generations with a population of 200 ships. In Brown and Mierzwicki 2004, a two-stage concept design strategy was proposed that they used a multi-objective optimisation and simplified risk event approach for concept exploration, and a more rigorous multi-disciplinary optimisation with uncertainty for concept development. The case study was based on a Mission Need Statement for an unmanned combat air vehicle (UCAV) carrier (CUVX).

Todd and Sen (Todd and Sen 1997) used a variant of MOGA for the pre-planning of container ship layouts (a large scale combinatorial problem). Four objectives were considered: maximize proximity of containers, minimize transverse center of gravity, minimize vertical center of gravity, and minimize unloads. Binary representation and roulette wheel selection with elitism based on non-dominance were used. They used the same algorithm in the shipyard plate cutting shop problem (Todd and Sen 1997; Todd and Sen 1998). Two objectives were considered: minimize make-span and minimize total penalty costs.

Peri and Campana (Peri and Campana 2003) proposed a multi-disciplinary design optimisation of a naval surface combat ship. The authors used an approach called multi-disciplinary design optimisation (MDO). It couples the analysis disciplines with numerical optimisation, and is a methodology for design of complex, coupled engineering systems. In order to formulate and solve the multi-disciplinary engineering problem of the hydrodynamic optimisation of a surface combatant, the David Taylor Model Basin (DTMB) model ship 5415 was used. Three objective functions were considered: two of them come from the solution of the partial differential equations (PDE) governing the motion of the ship in waves (heave and pitch response), solved in the frequency domain, and the third one comes from the solution of the PDE governing the steady motion of the fluid about the ship advancing in calm water (total resistance). The author proposed a recursive process with a careful selection of the samples to be placed in the design variable space. A subsequent refinement of the more promising solution is then undertaken, either applying a gradient method or shrinking the investigated space, re-centered on the more promising solution. With this approach, a subset of the Pareto optimal set is generated. The author used this method to avoid the high cost of evaluation.

Peri et al. (Peri and Campana 2005) used high-fidelity models and multi-objective global optimisation algorithms in simulation-based design. This work presented a simulation based design environment based on a Global Optimisation (GO) algorithm for the solution of optimum design problems. The procedure, illustrated in the framework of a multi-objective ship design optimisation problem, make use of high fidelity, computational models with expensive CPU time, including a free surface capturing RANSE solver. The use of GO prevented the optimizer to be trapped into local minima. The optimisation was composed of global and local phases. In the global stage of the search, a few computationally expensive simulations are needed for creating surrogate models of the objective functions. Tentative design, created to explore the design variable space, is evaluated with the inexpensive analytical approximations. The more promising designs were clustered, then locally minimized and eventually verified with high-fidelity simulations. New exact values were used to improve the meta models and repeated cycles of the algorithm were performed. A decision making strategy was finally adopted to select the more promising design.

Ölçer (Ölçer 2008) proposed a hybrid approach for multi-objective optimisation problems in ship design and shipping. In his study, software Frontier was employed to perform the optimisation via MOGA. The optimisation procedure in his work is explained in Figure 2.13.

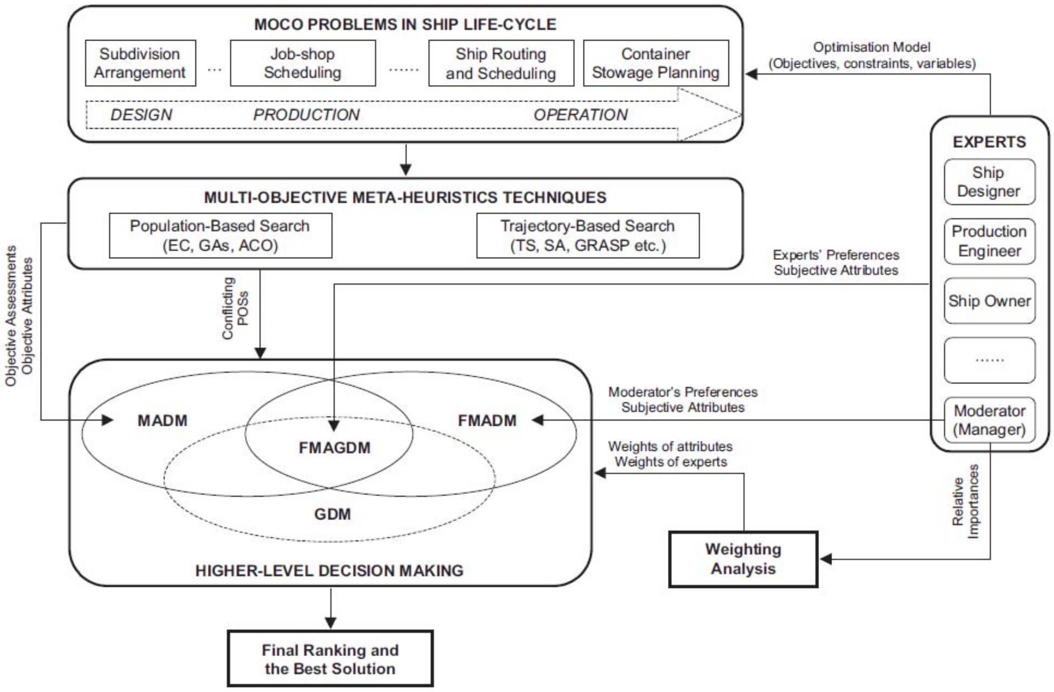


Figure2. 13 Global view for MOCO problems in ship design and shipping (Ölçer 2008)

Boulougouris and Papanikolaou (Boulougouris and Papanikolaou 2008) introduced a multi-objective optimisation of a floating LNG terminal. The software Frontier with MOGA was employed. The paper presented a comprehensive multi-objective hydrodynamic optimisation procedure and its application on the early design of a floating liquefied natural gas (LNG) terminal for improved seakeeping and wave attenuation characteristics on its lee side. Genetic algorithms were used to find the Pareto optima designs followed by multi-objective decision making procedures to select the optimum designs among them. The paper addressed the definition of the relevant optimisation problem and the set-up and interfaces of relevant software tools; presented results of systematic studies with respect to the minimization of the motion responses and wave elevation on the leeward side of free-floating terminals; and concluded with analysis and critical review of the obtained results and their impact on terminal design.

2.5.2 Multi-objective particle swarm optimisation

The Particle Swarm Optimisation is a global optimisation algorithm and described as sociologically inspired. It was first proposed by Kennedy and Eberhard in 1995 (Kennedy and Eberhart 1995). In the PSO algorithm, the candidate solution is the particle position in search space. Every particle is structured with two parameters: position and velocity, then the particle searches the solution space by updating the position and velocity. There are two best positions in PSO. First one is Pbest, which represents the best position that the particle itself can reach; the other is Gbest, which is the best position in the whole swarm.

The PSO can be described by the following equation,

$$\begin{aligned} v_{id}^{n+1} &= \omega v_{id}^{n} + c_1 r_1^n (p_{id}^n - x_{id}^n) + c_2 r_2^n (p_{gd}^n - x_{id}^n); \\ x_{id}^{n+1} &= x_{id}^n + \chi v_{id}^{n+1}; \end{aligned} \tag{2.1}$$

where x_{id}^n is the position of particle i, in n-th iteration and d dimension; v_{id}^n is the velocity of particle i, in n-th iteration and d dimension; p_{id}^n as Pbest is the best position of particle reached and p_{gd}^n as Gbest is the best position in current swarm; c_1 and c_2 are two coefficients; r_1 and r_2 are two random numbers with the range[0,1]; ω is the inertia weight and χ is constriction factor.

PSO has been proven as a simple but effective algorithm in single objective optimisation and multi-objective research using PSO has rapidly developed in recent years. Since the first extension was proposed in 1999, many different multi-objective PSOs have been presented (Margarita and Coello 2006) and some of these approaches are briefly given below.

Aggregating approaches: They combine all the objectives of the problem into a single objective and three types of aggregating functions are adopted: conventional linear aggregating function, dynamic aggregating function and the Bang-bang weighted aggregation approach (Jin, Olhofer et al. 2001), (Parsopoulos and Vrahatis 2002), (Baumgartner, Magele et al. 2004)

Lexicographic ordering: In this algorithm, which was introduced by Hu and Eberh (Hu and Eberhart 2002), only one objective is optimized at a time using lexicographic ordering Schema.

Sub-Population approaches: These approaches involve the use of several subpopulations as single-objective optimizers. Then, the subpopulations somehow exchange information or recombine amongst themselves, aiming to produce trade-offs amongst the different solutions, which were previously generated for the objectives that were individually optimised. Parsopoulos et al. (Parsopoulos, Tasoulis et al. 2004), Chow et al. (Chow and Tsui 2004) and Zheng et al. (Zheng and Liu 2007) used this approach.

Pareto-based approaches: These approaches use leader selection techniques based on Pareto dominance and references include Moore and Chapman (Moore and Chapman 1999), Ray and Liew (Ray and Liew 2002), Fieldsend and Singh (Fieldsend and Singh 2002), Coello et al (Coello and Lechuga 2002) and (Coello, Pulido et al. 2004) and Li (Li 2003).

Some of the other approaches, which use different techniques such as ***Combined Approach*** and ***MaxMin approach,*** can also be found in literature.

In PSO area in ship design, Pinto et al. (Pinto, Peri et al. 2007) emphasized the importance of initial points' configuration and addressed some preliminary aspects of global convergence of PSO towards stationary points. They presented a deterministic method for multi-PSO and applied the method to the multi-objective (two objectives) on sea keeping of container ship problem.

From the review, it can be seen that the multi-objective research generates more and more interest. In many GAs tested, the number of objectives is increased together with increasing constraints and as a result, the application environment becomes more complex. This naturally requires more sophisticated approaches that will enhance ship design.

2.6 Discussion

In this chapter, the technologies used in learning based design decision system are reviewed. Through the review, it is clear that the knowledge development is an emerging direction for design support system. The agent based system framework provides the great advantages for new system and it is a developing trend for new knowledge system. However, autonomous agent needs knowledge support and this raises the learning problem. Learning function is critically important for agent based design support system.

The machine learning is very rare in ship design and the most approaches are used via data mining software. A special method for ship design should be developed to improve the learning factors in the system, while for multi-objective optimisation, the new approaches should be tested. This is the main basis of the research presented in this thesis.

Chapter 3

Learning Based Ship Design Decision Support System

3.1 Introduction

Based on the critical review of the current expert system and machine learning technology, the decision based design concept and multi-agent system are employed to constitute the framework of this learning based ship design decision support system. There are several machine learning approaches being integrated to build the systematic knowledge learning method in this study while the whole system will be designed as distributed system for multi-tasks according to ship design and manufacture practice. The whole ship design work should be seen as the knowledge based multi-criteria fuzzy decision-making process.

The critical problem in the marine design and manufacturing, which the naval architects face, is the interlacing relationships of complex knowledge and experience with new design. The more challenging issue is that this relationship exists in the concurrent environment. The proposed system aims to solve this problem via building an efficient and feasible integrated approach to find, store and reuse the relationships intelligently for supporting the ship design in a dynamic environment. The different learning approaches are introduced, analyzed and developed to solve different problems and finally an integrated systematic methodology is proposed for ship design.

3.2 Framework of proposed system

Taking account of the natural complexity of ship design process, the decision support systems for ship design usually are very complex, which makes these systems very difficult to use. Simultaneously, most of these systems were organized linearly, which is not suitable for the concurrent engineering characteristics of modern ship design and manufacturing. What is more, previous expert systems normally need the special system designers to maintain and update new knowledge into the systems, which make maintenance hard and very expensive. All of above problems seriously restrict the application and development of ship design expert system. This study is utilising learning theory and multi-agent technology combining with the research based on previous work (Turkmen 2005). The whole work of this thesis attempts to develop a simple but effective system to develop and use in a very friendly way during the ship design process with the prior work being well integrated in this system. This system developed is oriented to concurrent engineering on the foundation of multi-agent system for satisfying the requirements of agile manufacturing. The key technique of this system is realizing the system self-update using machine learning methods, which greatly reduce the maintenance and developing cost of the system. All of these advantages help this ship design decision support system to move into a new intelligent age. The research emphasis and blueprint are described in detail as follows.

3.2.1 Research emphasis

The study puts the research emphasis on four aspects, which are the learning library building， new optimisation approach, real-time learning and learning based decision making.

First of all, every learning method, which is selected in this system, should be easy to understand clearly for both the system designers and the system users. This system is developed from the decision based design theory. The designer should take the role of the final decision maker while the support system takes charge of providing the full support for the designers. So the support system should provide clear and simple solutions to naval architects and all of these solutions should be clearly explained. This is the first key point of the support system: clear, simple and easily explained.

The second aspect is the experience of sharing and self-learning. Most of time, the design and production engineering highly depend on the personal experience of designers. There are two main problems in this situation. One problem is that this kind of design and production are too sensitive as the quality of design and production is very unstable and changes greatly according to the ability of designer. Especially, when there is no matured knowledge and experience, the absence of guidance will critically reduce the quality of final decision under uncertain design environment. The other problem is that the experience can not be updated automatically. Currently, most of experience is controlled by personal designers. The expert system has to continue introducing new experience via system designers which makes the maintenance of expert system very complex and expensive. This study attempts to find a method to automatically obtain experience from prior design case and then transfer the knowledge in order to guide the next design. The study in this aspect includes two parts. First part is collecting the experience from prior case and the second part is controlling the design process via real-time self-learning.

The third aspect is building an integrated optimisation method with learning function. This research takes most of the ship design process into account as the optimisation problem. In other words, the design process is multi-objective optimisation based decision making. So building a new optimisation approach with good inter-learning ability and simple parameters setting is necessary for this system.

The last aspect is the intelligent decision making method. This system accepts the decision based design concept, so the decision making method is the core of the system. The traditional methods need specialists to evaluate the optimisation results. But the high level specialists are often difficult to find during the design process. An experience based virtual committee can help designers to evaluate the designs via prior experience and improve the robustness of whole system.

3.2.2 Framework of proposed system

The framework of whole system which is shown in Figure 3.1, is developed according to multi-agent system theory and concurrent engineering concept. The single ship design and manufacture is still organized via linear mode but when several ships are designed and produced, the system can operate parallel to design and manufacture.

As shown in Figure 3.1, the framework of proposed system is divided into three parts. The first part, which locates at the left area above red line, is Ship Design Learning Library. The aim of this part is to provide strong support for experience storing and sharing. The sub-system of this part is built on multi-agent theory when the data is stored via XML. The development of this part is a new hybrid data storing and sharing mode when the relationship and data are operated together.

The second part which is the right area above red line, is the agent society. The reason of creating this agent society is aiming to fit the concurrent character of ship design and manufacture in real world. In this study, the research aims to improve the comprehensive ability but not single ship design. In real world, it is impossible that a design system just provide a service for a single ship and then, begin the next new design. For this aim, the proposed system is designed to distributed decision support system. The agent society does not specially belong to single ship design process but is public to all ship design process. In the agent society, the agents are classified to different agent group according to function. In Figure 3.1, three agent groups are provided as example. Every time, the ship design process gives a sign to require agent and this sign will be analyzed and redirected to agent group. The agent group will select an appropriate agent to perform this task. This part is the main research emphasis.

The third part is below the red line. This part is the application of proposed system. The design task 1 and design task 2, which are processed at the same time, are provided as the example of concurrent engineering. From Figure 3.1, it can be seen that the steps of design are the same for these two tasks. The green line express that every task has to draw the experience from SDLL. The yellow lines mean both tasks can apply for the same agent to do the same work at the same time. Based on the assumption that the task 2 is quicker than task1, blue lines clearly explain that different task can operate different step via applying different agent.

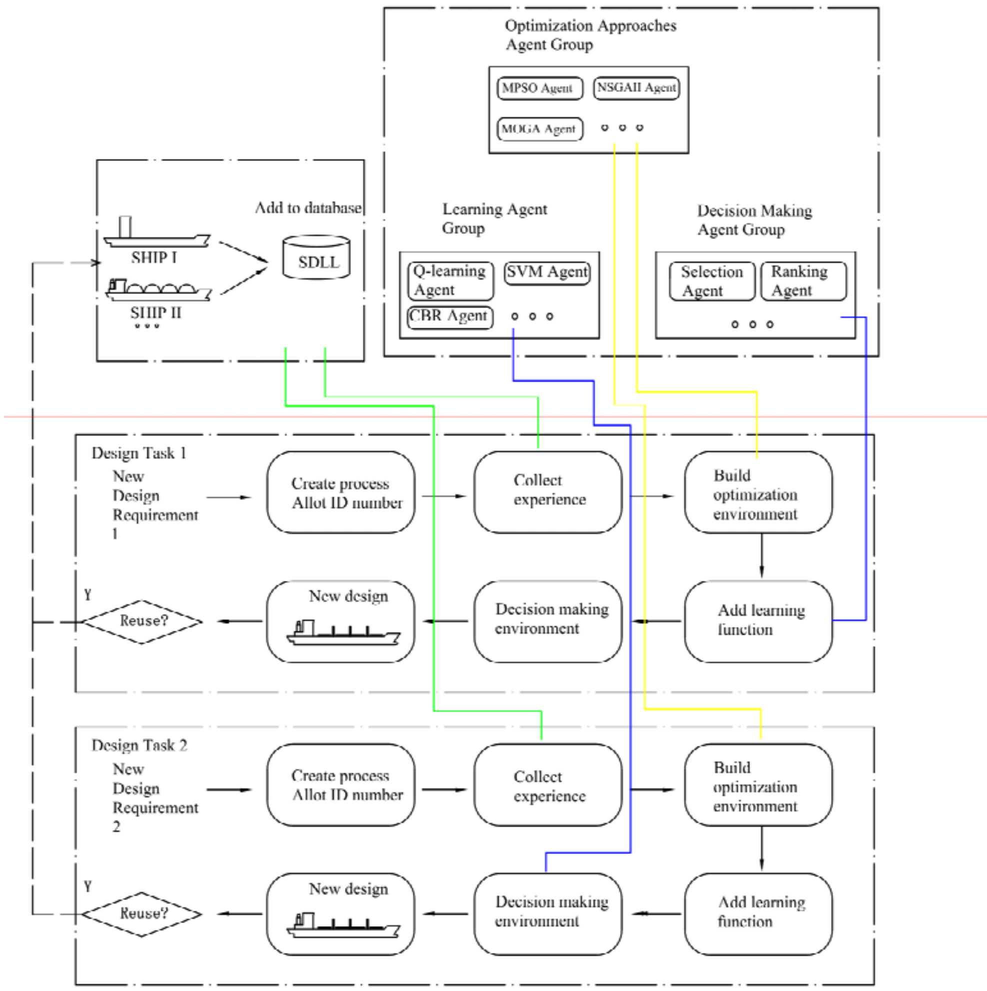


Figure 3. 1 The framework of learning based ship design decision system

3.3 Approach adopted and developed in this research work

This research focuses on employing and developing state-of-the-art machine learning, optimisation and decision making methods to create an integrated multi-agent ship design decision system. In order to develop this system, four new methods in different research fields are developed and two new practical applications are performed (shown in Figure 3.2). These new methods and applications lead the ship design work into artificial intelligence design decision system environment, which will provide more feasible design space and freedom design concepts under the guidance of self-learning experience.

The emphasis of this study has been put on both theory and practice. The employed and developed approaches in this research consist of two basic actions, which are vital in achieving the goal of this thesis. They are as follows: mathematic analyses and practical evaluation. The mathematical analyses action presents the research problems, builds the mathematical model and creates novel or integrated algorithms. The practical evaluation action provides practical problem solving framework and presents the application of real case study

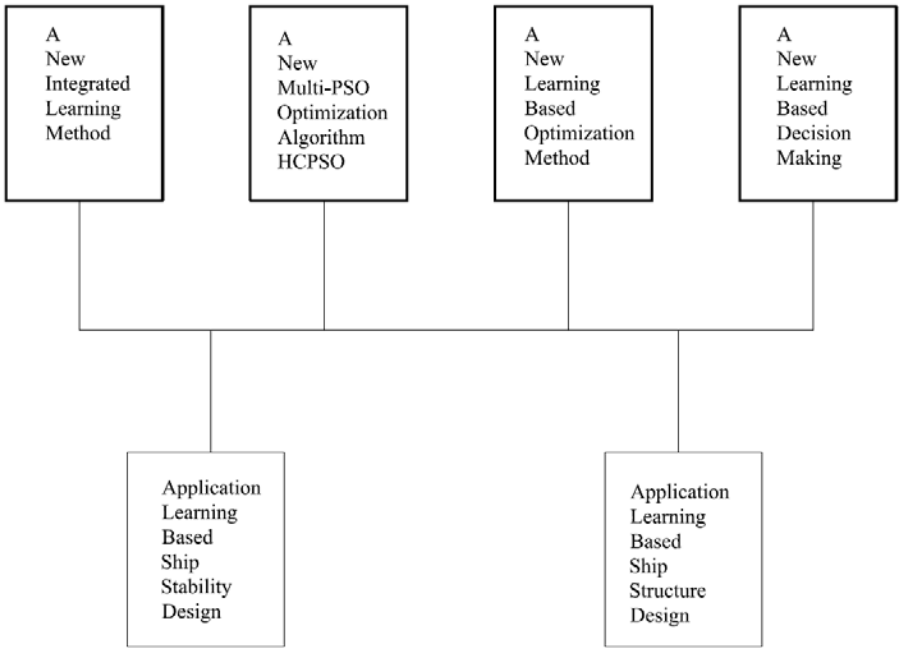


Figure 3. 2 Developed approaches in this study

The first method developed in this study is "*A New Integrated Learning Method for Ship Design Data*" as shown in Figure 3.2. The reason of developing this method is the storing model of the traditional drawings can not satisfy the requirements of new development of ship design pattern and has become one of the bottleneck problems. How to use new information technology of data operation to replace old data storing has been one of the most difficult problems for ship design. The aim of developing this method is to create a new data storing model of ship design for quick retrieving via assistance of learning algorithms. This new approach, which will be presented in chapter four, integrates two well established machine learning approaches: decision tree and case-based reasoning. For solving this problem, a new integrated learning method, which aims to find, store and reuse the relationship of every design data instead of simply keeping the data, is created. In other words, the new method can automatically retrieve the results using relationship of data and abandon old manual retrieval. At the same time, this method pays attention to improving the retrieval speed. In order to achieve this objective, the new method adopts different approaches to deal with different data. It is noteworthy to highlight that this method is sustainable which means the relationships, found via previous ships' data, can be revised or enhanced by newly added ships. Logically, this approach is the foundation of whole new design decision support system and it provides an excellent method of gaining experience from previous designs.

The second method is "*A New Multi-PSO Optimisation Method—HCPSO*" as shown in Figure 3.2. The reason of creating this new optimisation method is to provide the foundation for learning based optimisation. Because the ship design process in this study is seen as optimisation process, for better application of machine learning, the optimisation approach in this system should have two essential attributes: internal learning ability and simple parameters setting. The internal learning ability means that the new system need not to prepare special internal learning function especially for algorithm itself. The simple parameter setting means that the system can control the optimisation really well without too complex parameters setting. This new HCPSO approach has excellent attributes to satisfy these requirements with very good performance on optimisation ability proven in chapter 5.

The third method is "*A new learning based optimisation method*" as shown in Figure 3.2. It greatly improves the external learning of this system in real time learning mode. When the first and second methods in Figure 3.2 offer the abundant experience and advanced optimisation algorithm, how to combine them together to construct a high-performance intelligent collaborative learning system becomes primary problem.

This approach can capture the change of optimisation environment and provide guidance to optimisation. From the ship design optimisation point of view, the bottleneck problem in current application environment is the cost of time. In order to solve this problem in an effective way is to make the system more intelligent to assist the optimisation. This learning based optimisation can reduce the run time via intelligent guidance using the experience from both previous design and real time optimisation in chapter 6.

The last method is "*A New Learning Based Decision Making Method*" as shown in Figure 3.2. When the optimisation is finished, the system needs to select the final solution for designers. Most traditional decision making methods depend on human to make an evaluation. This will make the designs to become sensitive and when lack of specialists, it will be very hard to make decision or the decision may not be the best one. This learning based decision making method rebuild previous work-FMADM via multi-agent theory. This rebuilding work transfers traditional FMADM from semi-automatic to automatic, which enhances the robustness of the system. Then Support Vector Machine (SVM) method is introduced to construct the virtual committee which can learn human decision experience under the direction of virtual technology manager which is also built via SVM. This method can assist the designers significantly to make a good decision after optimisation. The theory and practice of "A New Learning Based Decision making Method" are explained in chapter 7.

Two application cases shown in Figure 3.2 are very important research problems in ship design. The first application, which is explained in chapter 8, belongs to ship design which focuses on stability. The subdivision optimisation of chemical tanker according to new SOLAS is selected as a case study. The third party naval architecture software NAPA is employed to realize the simulation and calculation. The second application in chapter 8 belongs to ship structure design and optimisation of mid-ship of bulk carrier. The Common Structural Rules (CSR) of International Association of Classification Societies (IACS) is selected as calculation basis. The Finite Element Analysis (FEA) software ABAQUS is utilised to assist the application.

3.4 Discussion

This chapter gives the research direction of the study. The system keeps the structure of multi-agent system and put the emphasis on developing the learning function of the system. The selection of the learning approaches follows the principle of simplification and effectiveness. At the same time, the construction of learning function is independent from other parts of the system. The learning function can be easy to update, which can help to better support the whole design process for realising a quick design and decision making.

The adopted and developed approaches in this research work are concluded and presented in this chapter. The logical relationship of these approaches is analysis together with the reasons and aims of employing and developing these approaches. The original contributions of the author are also briefly introduced.

Chapter 4

Data Mining in Ship Design

4.1 Introduction

Data mining is described as the process of extracting hidden patterns from data. The application of data mining is developed and extended in engineering field very rapidly in recent years with the improvement of statistics and computer science. At the same time, ship design suffers the problem of massive data, as with the development of computer application, naval architect encounters more and more unorganized data. As the traditional method of storing information in shipyard, the drawings used to be collected manually. However, in today's shipbuilding for current situation, the data contains many kinds of information which greatly exceeds beyond the range of drawings. What is more, the designers need quick control on both the explicit and implicit knowledge gained from previous experience. The original manual retrieval approach can not suit the development of ship design. All of these raise a new problem: how to manage the information and draw knowledge directly from database to provide an experience for the next ship design.

Data mining and machine learning present a new idea for solving above ship design problem. The suitable learning approach should have the ability to help the design support system to deal with the mass data and draw knowledge automatically from these data. But up to now, there has not been a general approach of machine learning, which can solve all the problems of data mining, especially for the application in ship design and production. In this study, a systematic learning method for sharing past experience in concept design of ships is presented to solve the complex ship data problems. Ship design learning library (SDLL), as a new concept database is introduced, while decision tree and case based reasoning approaches are selected to construct and run this new database. The theory and algorithm implementation is explained and a case is introduced for evaluating the method in this chapter.

4.2 Background and aim of this chapter

4.2.1 Background of data mining

Data mining is not a new concept but has a complete new development due to the new technology of statistics and computer science in recent years. The humans have analysed data to find pattern for centuries. At the beginning, the operation of data mining was processed manually, so the efficiency was very low. At the same time, the theory of statistics was not mature, which also constrained the application of data mining. In this progress over the years, many useful methods including Bayes theorem and regression analysis were developed, but overall, the development of data mining was very slow. Since 1950s, the development of computer technology and statistics theory has given a new life to the data mining. The research and application of data mining began to make a rapid development and now, data mining has been one of the important technologies for engineering.

"*Data mining is an iterative process within which progress is defined by discovery, through either automatic or manual methods. Data mining is the search for new, valuable, and nontrivial information in large volumes of data.*" (Kantardzic 2003). In this study, the data mining is defined as an effective method to find the implicit relationships of ship data via the pattern analyses. The modern engineering is based on the first-principle models. The normal process of engineering is that building a basic scientific model, and then building up and extending to various applications. The problem of ship engineering is that many ship engineering problems are too complex to be mathematically formalized. So the data should be analyzed to find the relationships and build an experienced model. "*Thus there is currently a paradigm shift from classical modelling and analyses based on first principles to developing models and the corresponding analyses directly from data*." (Kantardzic 2003). In other words, the system analyses the data and builds an experienced model via the value knowledge mining from existed ship data for improving the next practical design.

The task of data mining in ship engineering is analyzing the prior ship data and giving a predictive model. In recent years, data mining has been widely in use in engineering and developed different fields to build model. Normally, there are four main aspects of data mining application in engineering to help building a model.

Classification: allocation of data to the predefined groups. As massive amount of data is coming from different parts and different stages in ship design and production, the user needs to be very clear which data belongs to which part. For example, in ship design industry, the design department always makes the concurrent design with many ships. Suppose that there are three different main productions of ships: container ships, bulk carrier and LNG ship. When new data come, the designers need to give an accurate classification that which corresponding ship type the data should belong to can be identified easily. Let us assume in this example that this data belongs to container ships and it is the information of deck depth. The next time, when the new design task needs to design deck depth for a containership, the system will automatically provide this information to help the designers as experience. In other word, this classified data can be used directly for the same type of ship.

Clustering: the algorithm will try to group similar items together. In most of the time, especially in database building period, there is no predefined group which can be selected. Allocating the data to a specified group is difficult. So it needs the algorithm to provide similar data together to form a new group. For example, when a data about ship's length is classified as container ship as above example and if the system already has 100 examples, the system, which has now 101 vessels, will classify these 101 examples into two categories. Here, suppose the 100 meters length is the intermediate value, so the system will have two groups: less than or equal to 100 meters and more than 100 meters. *It is* noteworthy that this classification is automatic and some times, it needs designers to make a revision. In this example, the designer can modify this classification length limit of 100 meters, to different values according to different rules and regulations.

Regression: Regression in data mining means the system will find a function to express the data with the appropriate model. For simple data sets, if the system can find a function to express the data clearly, it will greatly improve the utilization of the data.

Association rule learning: It finds the relationships among variables. When the data is complex, it is difficult to use simple expression to describe the data model. So finding the explicit and implicit relationship of variables is favourable for utilization of data. This is the main application of data mining in ship design. In the environment of fast changing marine business, designers have to face large amount of data and to provide quick response. For example, in a ship design, length, breadth, depth etc. are given and the speed range is to be identified. The data mining should conclude the association rule among these factors for the designers and give corresponding speed range according to new input factors using knowledge derived from prior ship design cases.

There are other applications besides these four areas. But in this study, the application of data mining will focus on these four areas: classification, clustering, regression and association rule learning in ship design.

4.2.2 Aim of the application of data mining in ship design

The aim of the application of data mining in ship design is to build an intelligent learning based ship design learning library (SDLL), which can find the explicit and implicit relationships among data and provide these to the next practical design as the experience to improve the design quality. In other words, the SDLL has the ability of updating itself and every time, it can provide both the data and the relationships which come from the data to help to create better design. As part of the data mining, the following problems are studied:

- ✓ The decision tree and its application in SDLL;
- ✓ Case based reasoning and its application in SDLL;
- ✓ A hybrid algorithm for SDLL building;

4.3 Ship Design Learning Library (SDLL)

Ship Design Learning Library (SDLL) is a new concept database with the learning ability and is created for learning from complex ship design data. The word 'Library' is used here to replace 'database' because this system is not merely storing the data but exploring the self-learning, just like the library storing knowledge. This new database can find relationships among the data and update the new knowledge together while deleting the overdue knowledge. When a new problem is encountered, it can give all valid, accurate and timely information and experience drawing from pervious design cases to assist the designers. So SDLL is an intelligent database with the ability of self-learning for ship design.

The reason of building new type database for this system is multiplicity. First of all, the old ship information sharing mode obviously lags behind the requirements of modern ship design. In most of the time, the ship designers need to have an insight into the relationships behind data instead of holding massive amount of data. So the relationships among the data are more useful for designers. The second one is the type of stored data. As usual, the ship design agency always merely keeps the drawings but spurn other useful information. With the development of naval architecture, the information required in design process has been far beyond drawings. The storing and managing data is a challenge for designers. The third one is the utilization of knowledge. The development of marine marketing requires the designers to find more information to improve the design work and further improve the competitiveness. The optimisation is a good tool for improving the design but the current optimisation costs too much time. So it needs the knowledge to provide the good direction for guidance. The traditional knowledge to be shared from books often can not match the development of practical design, so a new database with self-learning and auto-updating function is necessary for ship design.

The learning ability of SDLL is realized via two learning approaches: decision tree and case-based reasoning. The decision tree mainly operates numerical rules while the case-based reasoning is employed to deal with linguistic rules.

The building and use of SDLL are not independent; on the contrary, they are closely related in practice and have no obvious boundaries.

The building part also can be seen as the training part. In SDLL, the learning methods are both supervised learning, which means that there are ‘teachers’ to give the instructions about learning process. In other word, past cases give the guidance to find the solution of new problem.

In this phase, the first important aspect is the data format. In previous work, the XML format is employed and for the continuous development, the new SDLL still accept the XML format. The XML format has many advantages: simple structure, easy to debug, online application etc. But the XML is not mature enough in current situation in large database. Therefore the learning module is designed to be independent from the database and can be embedded into any other module including module written via other database languages.

The second important aspect is the learning ability realization, which is one of the difficult parts of SDLL. The learning ability of SDLL means not only learning before the application but also learning during the application process. And it does not only give all the mature rules and regulations before the new application but also makes a concrete analysis of well established concrete conditions. For different learning problems, SDLL employs different strategies. For a numerical problem, SDLL will adopt the decision tree approaches to search the relationships and to store these relationships in the database. For linguistic properties of cases, the properties will be directly put into the database without any other operations. In other word, the linguistic properties will not be processed until there is a requirement given by new application.

This raises two problems. The first one is how to link the linguistic attributes to other attributes. The second one is how to retrieve the linguistic attribute.

For the first one, SDLL uses uniform index number, which means once a new case comes into the SDLL, a unique ID number, which is composed of the time (day, month, year) plus four numbers given in SDLL. It is noteworthy that the case will be given a sequence number when the decision tree finishes classification. But the new sequence number is just for storing in sub-database and will not be treated as retrieve number. So if the linguistic attribute is not treated as a retrieval attribute, it will be given as a part of the result via ID number.

For the second problem, the linguistic attribute is one of the retrieval attributes. It is more complex. Normally, in the retrieval process, linguistic attribute is coupled with other attributes. If CBR is used, one important problem is how to measure the distance between neighbour cases as numerical attributes. SDLL accepts man-machine cooperation idea to deal with this problem. The system divides the linguistic attributes into two classes. One is precision matching and the other is fuzzy matching.

For precision matching, the designer can directly select total matching cases and then measure the distance of other attributes to decide the best solution. For example, for a stability design, the system wants to use IMO A256 as design standard. In this situation, measuring the distance has no practical significance or even may be misleading. So the system just searches the case with IMO A256.

The last important aspect is the structure of SDLL. As presented in Figure 4.1, which gives the structure of SDLL, there are three libraries in SDLL. For linguistic attribute library, there are only two parts: ID number and linguistic attributes. For numerical attribute library, the relationships among the data are analysed and stored in the relationship library. The rules and regulations library stores the standards coming from the classification society, IMO, flag nations etc. and uses these to correct the linguistic and numerical attribute library.

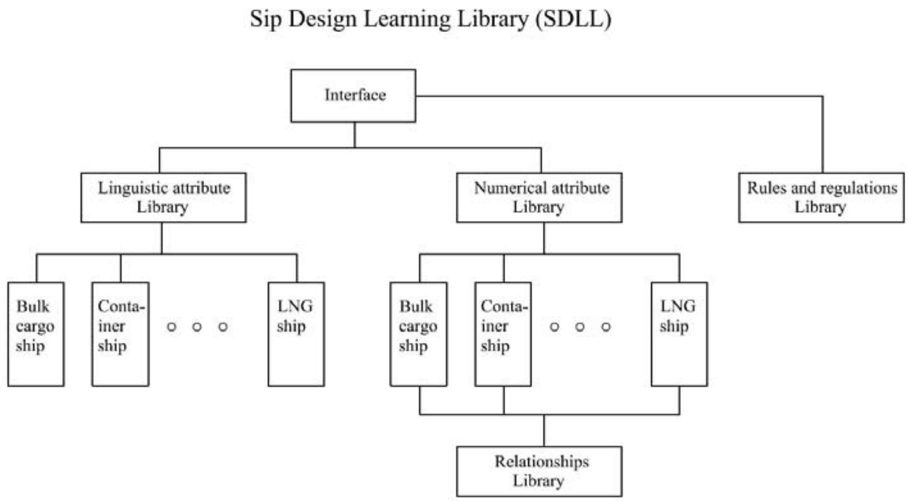


Figure 4. 1 The structure of SDLL

4.4 An integrated learning method for building SDLL

According to the characteristics of SDLL, the main task of SDLL is finding and storing the relationships behind the data. As the designers need to obtain the useful data from massive data of ship design. In this section, a new integrated data mining method will be put forward to deal effectively with the massive data.

Before selecting an appropriate approach, the composition of data in ship design should be taken into account. The main data can be divided into two categories:

numerical and linguistic. Then the numerical data can further be divided into drawings, tables, etc. In this research, without loss of generality, the data is classified under three categories: numerical data, drawings and linguistic data. Because the drawings can not directly be analysed by current technology, the objective data is limited to numerical and linguistic data. Correspondingly, the data mining approaches should also be considered for dealing with these two categories data.

A hybrid learning method is created to deal with this task. The decision tree is appropriate method for the numerical data while CBR is employed for the linguistic data. The detailed method has been shown in Figure 4.2.

As shown in Figure 4.2, the learning method is composed of two approaches: decision tree and CBR. The applications of these two approaches belong to different types of data. The decision tree will find the relationships behind the numerical data during the training time, and at the same time, the CBR make an analysis of linguistic data after receiving the new requirements.

At the beginning, a data classifier is created to different types of data. Because this study will not process the drawings and the drawings are directly store in the data warehouse, the categories are limited to two types: numerical data and linguistic data.

After all the training data is put into database, the decision tree will classify the attributes firstly. The name and value are also separated in this step. Then the decision of what kind of decision tree should be created becomes the primary problem. Because every attribute can be used as root node, a simple and effective decision tree becomes the first objective. The system needs to make the preference bias clear, in other words, the system has to decide what kind of standard should be preferentially used in the process of building trees.

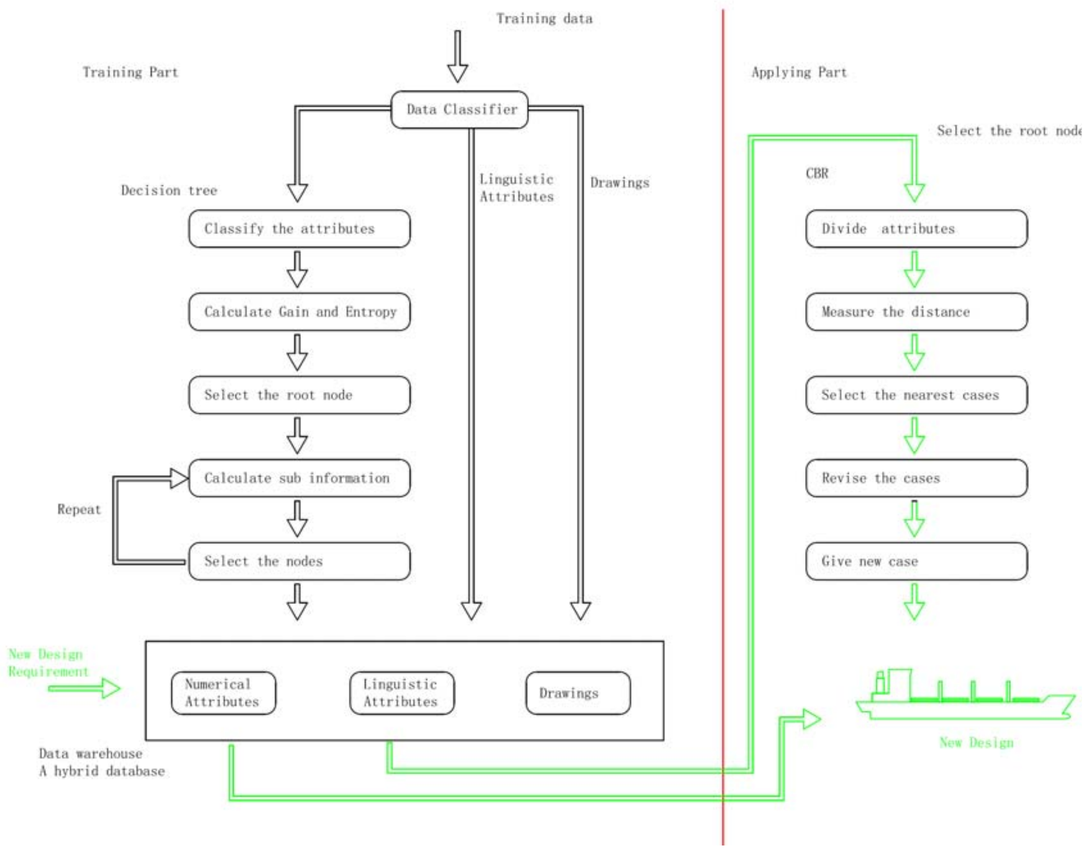


Figure 4. 2 An integrated learning method for building SDLL

For mass data, there could be many kinds of trees according to same data sets. How to build an effective tree becomes important. The Ockham's Razor is selected in this study as preference bias. The basic idea about Ockham's Razor is that the simplest explanation which is consistent with all observations is required as the best. This means the smallest decision tree that correctly classifies all of the training examples is the best. Finding the smallest decision tree is an NP (nondeterministic polynomial time)-Hard problem, so instead of constructing the absolute smallest tree which is consistent with all of the training examples, construct one that is relatively small.

After decisions about the principle of preference bias, the next problem is what method should be used to measure the difference among attributes of every case, which means how to make sure the best attribute for a given node. Most algorithms that have been developed for learning decision trees are variations of the core algorithm that employs a top-down, greedy search through the space of possible decision tree. This means the method will search the complete space of attributes.

For solving the above problems, an approach should be selected to decide the basic root node and following nodes of every lever. A method named Max-Gain, which is employed by decision tree algorithms, is also accepted in this integrated method. The Max-Gain comes from the information theory. The concept of Max-Gain is to choose the attribute that has the largest expected information gain. In other words, try to select the attribute that will result in the smallest expected size of the sub-trees rooted at its children. Information theory is used to estimate the size of the sub-trees rooted at each child for each possible attribute; that is, try each attribute, evaluate and pick the best one.

After obtaining the value of every attributes, the system will select the attribute with maximum gain value as the root node. Then the system will calculate the remaining attributes and select attribute from these remaining attributes with maximum gain value as the next lever node. The loop continues until the objective attribute is distinguished clearly. Until this step, the relationship of this objective attribute to others has been built. Then the next attribute will be analyzed, and after analysing all the attributes, the relationship net among attributes will be built.

This relationship net will be stored together with linguistic attribute and drawings in the data warehouse as presented in Figure 4.2. This warehouse is hybrid system combining the operated data and original data. The operated data means the discovered relationships of the data which has been stored and ready for use. In Figure 4.2, the data warehouse is divided into three parts for different data and in this study, the language XML is used to build the whole data warehouse.

For the application stages, the same data classifier of training part is employed to classify the new design requirement. The numerical attributes will directly be given to the relationship nets and the result will be obtained.

The linguistic attributes will active the CBR approach. Firstly, the linguistic attribute will be divided into several small parts for the next step. The detailed amount of parts is decided by the designer according to the concrete problem. For example, a speed can be divided into slow (<16 knots), normal (16 to 18 knots), good (18 to 22 knots), fast (>22 knots) for seagoing cargo tankers and also can be divided into slow(<8 knots), normal (8-12knots)，rapid (>12 knots) for inland ships. It is notified that when the standard of divisions is established, every case in the database will be divided. So the designers should take into account the whole design space and not to be absorbed by the design requirements. In the rent step, the distance between the requirement and the original case is measured and the nearest neighbours are identified. As the final step these cases are revised and by combining the result obtained via decision tree a new solution is provided.

4.4.1 Work flow of SDLL

While Figure 4.3 presents the work flow of SDLL, the detailed steps are given as follows:

Training part

Step 1: Read a new case, then transfer the data to SDLL format.

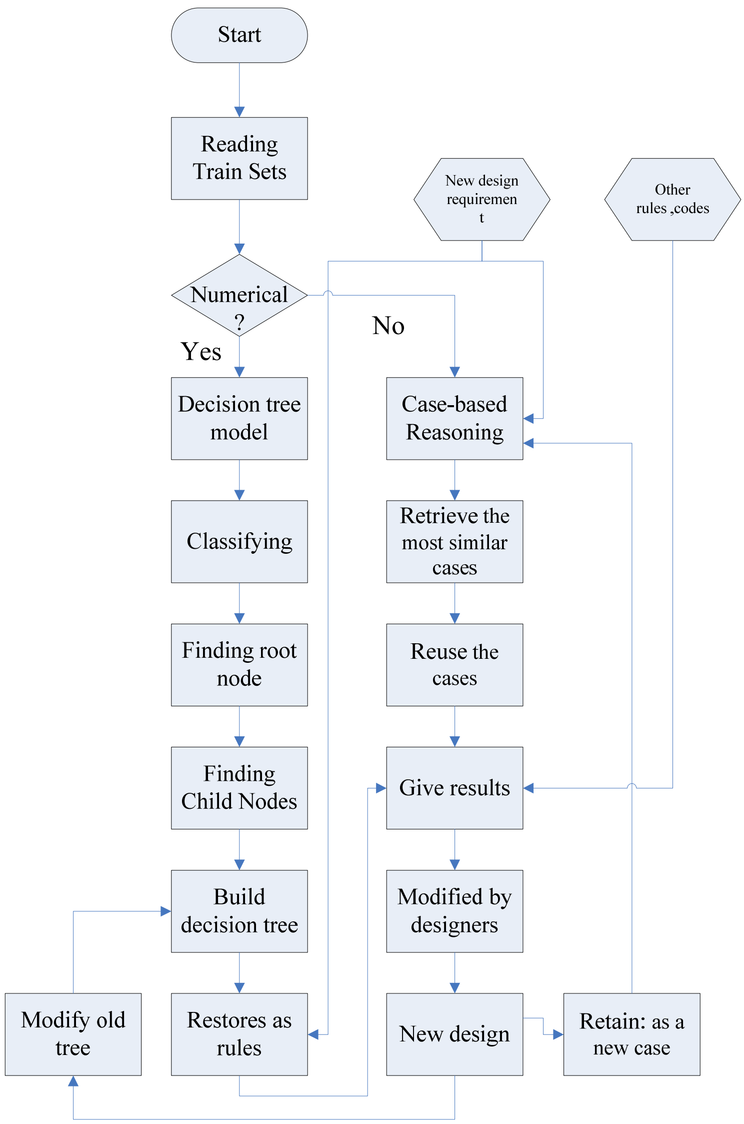


Figure 4. 3 The work flow of building Learning Library

Step 2: Classify the case attributes. If the attributes are linguistic, the attributes are directly transferred to case-based reasoning sub-database. Only ID numbers and linguistic attributes are stored;

Step 3: All other attributes are transferred to decision trees sub-database and the attributes of every case are re-sorted according to previous classification.

Step 4: When all the cases in the training set are put into the database, the decision tree method is called to process the data.

Step 5: Steps from 5 to step 8 include the decision tree inner work flow.

Considering set S as training examples, which consist of the subset P of positive examples and subset N of negative examples;

Calculate the entropy S via Equation 4.1. The definition of entropy and gain is explained in appendix A.

$$Entropy(S) = -(P\log_2 P) - (N\log_2 N) \tag{4.1}$$

where P=p/(p+n) and N=n/(p+n)

P: fraction of examples where S is positive;

N: fraction of examples where S is negative;

Step 6: Calculate information gain by using a given attribute via Equation 4.2, which measures the expected reduction in entropy. That is, measure the difference in the information content of a node and the information content after a node splits up the examples based on a selected attribute's possible values.

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|Sv|}{|S|} Entropy(Sv) \tag{4.2}$$

Where A is the set of all possible values for *attribute a*, and Sv is the subject of S for which *attribute a* has value v. The second term is the expected value of the entropy after S is partitioned using *attribute a*.

Step 7: Select the attribute with the highest normalized information gain, and create a decision node on the best attribute on this information gain. The best attribute is the attribute with the highest normalized information gain. This decision node will be the new sub-root node for the next time.

Step 8: Repeat on the sub-lists obtained by splitting, and add those nodes until all the cases are classified.

Step 9: Store the decision tree in the relationship library.

Running part

Step 1: The SDLL interface transfers the new design requirement to standard format.

Step 2: If the attribute is linguistic, enter linguistic attribute library else enter relationships library (go to Step 7).

Step 3: Judge whether it is a precision matching or a fuzzy matching. If it is a precision matching, directly search the cases with same attribute and measure the distance of other attributes. If it is a fuzzy matching, it needs designers to classify the linguistic attribute and transfer to numerical attribute.

Step 4: Calculate the distance between the cases with the pointed weights.

Step 5: Select the case and output.

Step 6: Ask the designers to revise the results and store as a new case in the library.

Step 7: Search the relationships library and give results.

Step 8: Ask the designers to revise the results and if necessary, revise the old decision tree.

4.4.2 Essence of data mining in SDLL

When SDLL is built successfully, after carefully training, it can be used in new design. The findings, which are given via machine learning method, are feasible in ship practical level. This section will investigate the essence of data mining in SDLL.

Ship design, as an engineering application, follows the general law of engineering science which is observing situation, building model, finding mathematical solutions, then exploring the general application. So ship design, in a certain degree, is seeking the mathematical solutions to all kind of ship related problems. But it is too complex to find all the solutions for all the problems in ship science; therefore, there are many empirical formulas which actually are fitted based on data from prior cases with correction via experimental data. With the development of marine industry, large numbers of new types of ships appear, for example, the container ships have many new types in recent years as the displacement increasing continuously. On the other hand, the corresponding study can not follow the market development. Both the empirical formulas and analytical theory including the study of flow fields, velocity fields etc. in ship design cannot follow the development of ship building. It also needs a very long time to transfer this research to knowledge and to teach designers. So the modern ship design needs to draw experience directly from the prior cases to rapid design in the constraints of rules and regulations, which are made by classification society, IMO etc. The machine learning and data mining can deal with these tasks effectively.

What kind of knowledge is mined from these cases? They can be seen as special rules and regulations in the current known attributes in specific cases. These relationships can be corrected via published rules and regulations. So they can be directly used in practice. What is more, many of these relationships can be transferred to general knowledge after research. It can also promote the development of ship science.

4.5 Discussion

This chapter proposes a systematic learning approach to build SDLL. The decision tree and CBR are employed to solve the learning problems in the SDLL. The theory and reason of selecting these two learning approaches are explained. The detailed workflow including both training and running parts is given.

This chapter introduced a new type learning based database for ship design work. The system can automatically learn from prior cases and provide help to the designers for improving the quality of the design. The decision tree is to be used for the first time for analysing the relationships behind ship data, while the CBR is to be used for the first time for linguistic attributes of ship data. Both of these approaches provide good performance for relevant data and demonstrate the power of method for utilising the past experience to develop high performance design.

Chapter 5

Multi-objective Optimisation and Multi-PSO in Ship Design

5.1 Introduction

As ship design is a systematic process, multi-objective optimisation in ship design has attracted a lot of interest in recent years. Many algorithms have been developed to solve the design and analyses problem in ship design, which is dynamic and complex concurrence engineering. The design process usually takes a long time to make the analysis of the problem. However current algorithms for GAs always need big population and steps, which make the time to complete the optimisation very long. While this problem requires faster algorithm, in order to be able to introduce learning function to the optimisation approaches, the optimisation algorithm should be as simple as possible. What is more, the algorithm should have the learning ability, in other words, the inner learning of the optimisation algorithm should be processed by the algorithm itself.

This chapter introduces a new ‘Hybrid Co-evolution based multi-objective Particle Swarm Optimisation’ (HCPSO) method to solve the above problem. The HCPSO combines co-evolution, game theory and ε-disturbance technique to develop an effective optimisation approach which performs remarkably well in the multi-agent system.

5.2 Background of multi-objective optimisation

The main aim of a general multi-objective optimisation problem (also called multiple criteria optimisation, multi-performance or vector optimisation) is to find the design variables that optimize a vector objective function ($F(Y) = \{f_1, \ldots, f_t\}$) over the feasible design space (Ölçer 2008). The objective functions are the quantities that the designer wishes to minimize, maximize, or attain a certain value. This problem can be formulated as follows:

Minimize: $F(Y) = \{f_1(Y), f_2(Y), \ldots, f_t(Y)\}$ (5.1)

Subject to:

p inequality constraints $g_\delta(y) \geq 0, \delta = 1, \ldots, p$

q equality constrains $h_\Phi(y) = 0, \Phi = 1, \ldots, q$

where $Y = [y_1, y_2, \ldots, y_n]$ is the vector of decision variables.
In multi-objective optimisation, the objectives are usually in conflict with each other. The aim of multi-objective optimisation is to find a solution which is acceptable to decision makers.

Design variables are the numerical quantities for which values are to be chosen in an optimisation problem. In most engineering applications, the design variable is controllable by designers according to factual problems. Design variables usually have maximum and minimum boundaries which can be treated as separate constraints.

There are various restrictions from the environment or resources (e.g., physical limitations, time restrictions, etc.), which must be satisfied for an acceptable solution in an optimisation problem. These restrictions are generally called constraints and they could be explicit or implicit.

In multi-objective optimisation, the aim is not just to find a single solution as global optimisation but to find good compromises (or "trade-offs"). Here Pareto Optimality is introduced.

For a multi-objective optimisation problem, any two solutions y_1 and y_2 can have one of the two possibilities: one dominates the other or none dominates the other. In a minimization problem, without loss of generality, a solution y_1 dominates y_2 if the following two conditions are satisfied:

$$\forall \gamma \in \{1,2,...,t\} : f_\gamma(y_1) \leq f_\gamma(y_2) \qquad (5.2)$$

$$\exists \lambda \in \{1,2,...,t\} : f_\lambda(y_1) \leq f_\lambda(y_2) \qquad (5.3)$$

If any of the above conditions are violated, the solution y_1 does not dominate the solution y_2. If y_1 dominates the solution y_2, y_1 is called the non-dominated solution. The solutions that are non-dominated within the entire search space are denoted as pareto-optimal and constitute the pareto-optimal set or pareto-optimal frontier.(Ölçer 2008)

5.3 Description of the proposed approach-HCPSO

The proposed approach combines co-evolution approach, Nash equilibrium and ε-disturbance technique to form a new improved hybrid approach. At the same time, agent-based structure is adopted for distributed synchronous cooperation.

5.3.1 Nash-optima in proposed method

In HCPSO, co-evolution approach is combined with Nash optima, which looks for Nash Equilibrium. For an optimisation problem with M number of objectives in an agent environment, M numbers of agent-groups are employed to optimise their own objective while changing the objective of other/remaining agent-groups. When no agent-group can improve its objective further, it means that the system has reached a state of equilibrium called Nash Equilibrium. In HCPSO, Co-evolution approach provides a public information sharing mechanism to improve the communication amongst agent-groups.

The reason for adopting the co-evolution approach here is that on the one hand, co-evolution approach can provide quicker search ability, and on the other hand, by using the co-evolution approach it is easier to perform simulation via a multi-agent system. In an engineering application, the approach can be studied in a multi-computer environment thereby reducing time via parallel calculations.

The Nash optima (Sefrioui and Periaux 2000)，(Holt and Roth 2004) is deployed in HCPSO because the information sharing in Nash optima is simple and effective. Furthermore, in Nash optima, one agent, which has one objective, receives the information from other agents, and this is a good model for the multi-agent system in concept and computer realization.

5.3.2 ε-disturbance in the proposed method

In the multi-objective PSO method, sometimes the algorithm will generate local optima value and the ε-disturbance technique is deployed to avoid it. When both Pbest and Gbest stand on a fixed value and the particle position is stagnant on boundary, a random ε-disturbance is needed to help the algorithm jump out of stagnation point. For a given step T, if stagnant step $t_s > T$, a random ε-disturbance is introduced to both Pbest and Gbest. For example, setting T=10 steps, the algorithm would check the value of Pbest and Gbest at every step. If both Pbest and Gbest keep the same values as the last step, the counter in algorithm will increase by one. In the next iteration, if both Pbest and Gbest still keep the same values, the counter will further increase by one, otherwise the counter returns to zero. When the counter is equal to 10, the algorithm will give a random ε-disturbance value to change Pbest and Gbest, so as to keep the particles moving.

5.3.3 The proposed method in study

For an optimisation problem with K design variables and M objectives, the algorithm would employ N particles to run optimisation and N should be multiples of M. Firstly, HCPSO divides N particles into M sub-swarm group according to objectives. Secondly, the design variables, K, are also divided into K_1, K_2, … K_m sub-groups according to objectives. This means every sub-swarm group has its own corresponding objectives and design variables. Then every sub-swarm group optimises relevant design variables according to relevant objectives in its own group. The optimised result would be sent to the "public board" as shared information. In the next step, the sub-swarm group would read other design variables and corresponding information from the public board as an update. At last, when a Nash Equilibrium is reached, the algorithm gives the final results. In every sub-swarm, in order to give enough pressure to push the solution space to Pareto space, particles in two generations are compared to update the position.

Algorithm HCPSO

The step-wise description of the proposed algorithm, HCPSO, is given below and shown in Figure 5.1:

1. initialize the population including position and velocity;

   (i) create public-board to store information; public-board is a simple database to store all particles' information;

   (ii) give the number of sub-swarms and size of each sub-swarm;

   (iii) divide population into sub-swarms;

2. every sub-swarm begins iteration; for example (m^{th} sub-swarm with (N/M) particles and K_m design variables);

3. read information from public-board and calculate fitness;

4. update Pbest via comparing individual position history according to m^{th} objective ;
5. non-dominated sorting via Pareto-based approach;
6. give pointed number of best ranking particles to public-board;
7. collect particles in step 6 from all sub-swarm to public-board and form Gbest pool;
8. every sub-swam randomly selects Gbest from pool;
9. compare Pbest and Gbest with past values to decide whether to introduce ε-disturbance or not;
10. calculate velocity and give limit velocity;
11. update and limit the position of particles;
12. combine last iteration particles with this iteration to form a new group with 2*(N/M) population and perform non-dominated sorting via Pareto-based approach;
13. select first (N/M) particles from new group with 2*(N/M) population in step12 and update position and velocity;
14. send corresponding information to public-board;
15. if every solution can not be improved, stop and if it can be improved, continue iteration;
16. output final results;

Sub-swarm partition

In HCPSO, the whole swarm is divided into different sub-swarms according to the number of objectives. If an optimisation problem has M objectives, the swarm is divided in to M sub-swarms. The relationship between variable and sub-swarm is very important. For special objective functions, detailed analysis of these relationships should be processed before partition. For example, a two objective problem: $f_1(x) = x_1, f_2(x) = g(x_2, x_3, ..., x_n)$; following the above principle, the swarm should be divided into two sub-swarms. The first choice of variables is allocated to the first sub-swarm which contains variable x_1 and second sub-swarm includes the variables from x_2 to x_n. The second choice is averaged out and every sub-swarm has half variables. In a particle application in engineering, when the relationship is unclear, the average distribution is recommended.

Information sharing mechanism

The information sharing mechanism is realised through public-board in sub-swarm optimisation. The public-board concept in HCPSO comes from a multi-agent system environment. For easy and efficient communication in an agent environment, a black board is used for sharing the public information. The same mechanism is used here, and the black board is called public board for public information sharing. In every sub-swarm, the particles optimise variables according to sub-swarm optimisation objective, and send the information including the optimised objective value, the best particle location etc. to public board in the system. In synchronization, the sub-swarm reads the information from the other sub-swarm optimisation as the information for the next step.

Pbest and Gbest updating

The Pbest updating in HCPSO adopts the updating inside sub-swarm while Gbest uses updating outside sub-swarm. For Pbest, the particle in n^{th} sub-swarm compares the corresponding objective fitness value between the current position and the previous best position. If the current position in the n^{th} objective fitness is better than the previous value, the Pbest is replaced by the current value or if the current value is not better, then the previous value is retained.

The Gbest selection uses Gbest pool in public board. Every sub-swarm provides its best value to the public board and constructs a Gbest pool and then HCPSO randomly selects a particle as Gbest position from Gbest pool.

Particle updating

In order to give enough pressure to Pareto front, an updating position approach is used to mutate every particle. In every sub-swarm, the system will compare the fitness in the last generation and the current generation (size of population is $2\times N/M$), then selects N/M position as this generation; some particles hold their own position and others will be replaced by the nearest neighbour.

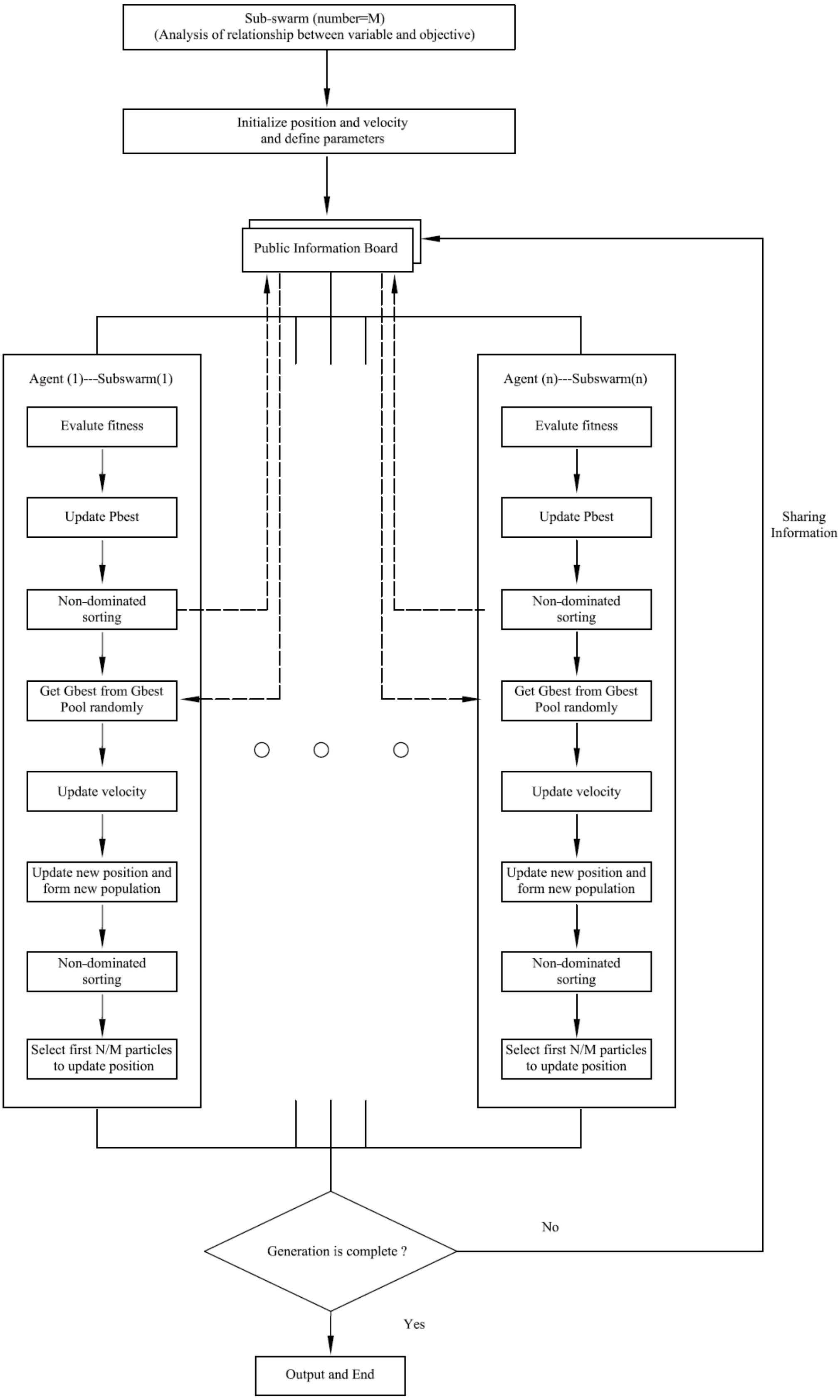


Figure 5. 1 Flow chart of proposed HCPSO algorithm.

5.4 Experiments

5.4. 1 Test Functions

Eight multi-objective test functions (Table 5.1), from the related literature, are selected to evaluate the proposed algorithm. Deb (Deb, Thiele et al. 2001) introduced the design of a multi-objective test problem for EAs (Evolutionary Algorithms) and the test problems in Table 5.1 are chosen following the principles of Deb. Test 1 is selected by Pinto et al. (Pinto, Peri et al. 2007) as a convex Pareto front. Test 2 is ZDT1 function in Zitzler et al. (Zitzler, Deb et al. 2000) with continuous Pareto front and a uniform distribution of solutions across the front. Test 3 is ZDT2 in which the Pareto front is not convex. Test 4 is obtained combining the test function of Test 2 and Test 3 (Jin, Olhofer et al. 2001) and explored by Pinto et al. (Pinto, Peri et al. 2007). Test 4 is neither purely convex nor purely non convex. Test 5 is ZTD3, which is difficult for a discontinuous Pareto front. Test 6 is ZTD4 and the difficulty of this problem is that the sheer number of multiple local Pareto-optimal fronts produces a large number of hurdles for an algorithm to converge to the global Pareto-optimal front.

Table 5.1 Two-objective test problems selected to evaluate HCPSO algorithm.

Name	Variables n	Range	Test Function
Test 1	30	[0,1]	$\begin{cases} f_1(x) = \frac{1}{n}\sum_{i=1}^{n} x_i^2, \\ f_2(x) = \frac{1}{n}\sum_{i=1}^{n}(x_i - 2)^2. \end{cases}$
Test 2	30	[0,1]	$\begin{cases} f_1(x) = x_1, f_2(x) = g(x)h(f_1(x), g(x)). \\ g(x_2,...,x_m) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i, \\ h(f_1, g) = 1 - \sqrt{f_1 / g}. \end{cases}$
Test 3	30	[0,1]	$\begin{cases} f_1(x) = x_1, f_2(x) = g(x)h(f_1(x), g(x)) \\ g(x_2,...,x_m) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i, \\ h(f_1, g) = 1 - (f_1 / g)^2. \end{cases}$
Test 4	30	[0,1]	$\begin{cases} f_1(x) = x_1, f_2(x) = g(x)h(f_1(x), g(x)) \\ g(x_2,...,x_m) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i, \\ h(f_1, g) = 1 - \sqrt[4]{f_1 / g} - (f_1 / g)^4. \end{cases}$
Test 5	30	[0,1]	$\begin{cases} f_1(x) = x_1, f_2(x) = g(x)h(f_1(x), g(x)) \\ g(x_2,...,x_m) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i, \\ h(f_1, g) = 1 - \sqrt{f_1 / g} - (f_1 / g)\sin(10\pi f_1). \end{cases}$
Test 6	10	$x_1 \in [0,1]$, $xi \in [-5,5]$, $i = 2,...30;$	$\begin{cases} f_1(x) = x_1, f_2(x) = g(x)h(f_1(x), g(x)) \\ g(x_2,...,x_m) = 1 + 10(n-1) + \sum_{i=2}^{n}(x_i^2 - 10\cos(4\pi x_i)), \\ h(f_1, g) = 1 - \sqrt{f_1 / g}. \end{cases}$

Two three-objective test functions are selected according to Deb (Deb, Thiele et al. 2001) in Table 5.2. Test 7 is DTLZ2 and is used here to investigate HCPSO's ability to scale up its performance in a large number of objectives. Test 8, as DTLZ4, is selected to test HCPSO's ability to maintain a good distribution of solutions.

Table 5.2 Three-objective test problems selected to evaluate HCPSO algorithm.

Name	Variables n	Range	Test Function
Test 7	12	[0,1]	$\begin{cases} f_1(x) = (1+g(x_M))\cos(x_1\pi/2)\cos(x_2\pi/2) \\ f_2(x) = (1+g(x_M))\cos(x_1\pi/2)\sin(x_2\pi/2) \\ f_3(x) = (1+g(x_M))\sin(x_1\pi/2) \\ g(x_M) = \sum_{x_i \in x_M} (x_i - 0.5)^2 \end{cases}$
Test 8	12	[0,1]	$\begin{cases} f_1(x) = (1+g(x_M))\cos(x_1^{100}\pi/2)\cos(x_2^{100}\pi/2) \\ f_2(x) = (1+g(x_M))\cos(x_1^{100}\pi/2)\sin(x_2^{100}\pi/2) \\ f_3(x) = (1+g(x_M))\sin(x_1^{100}\pi/2) \\ g(x_M) = \sum_{x_i \in x_M} (x_i - 0.5)^2 \end{cases}$

5.4.2 Parameters Setting

As a critical parameter for the PSO's convergence behaviour, ω is utilised to control the impact of prior velocities. Usually, a large ω facilitates global exploration while a small one tends to facilitate local exploration. The experimental results indicate that it is better to set the inertia to a large value initially, in order to promote global exploration of the search space, and gradually decrease it in order to obtain more refined solutions. The proper setting of parameters $c1$ and $c2$ may result in faster convergence and alleviation of local minima. The constriction factor χ controls the magnitude of the velocities, in a similar way to the *Vmax* parameter (Parsopoulos, 2002).

The standard PSO is the original algorithms provided by Kennedy and Eberhard (Kennedy and Eberhart 1995), which is similar to equation (2.1) in subsection **2.5.2** but without χ. The coefficient ω is equal to 1 and $c_1 = c_2 = 2.0$.

In order to compare the rate of convergence between standard PSO and its improved form, two test functions in the literature are selected (Shi and Eberhart 1999), (Eberhart and Shi 2000):

1. De Jong's function

$$f(x)=\sum_{1}^{n}x_i^2\ ,\ -5.12\le x_i\le 5.12 \tag{5.4}$$

Global minimum $f(x)=0$ is obtainable for $x_{i=0}, i=1,...,n.$

2. Griewangk's function

$$f(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^{\ 2}-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1,\ -600\le xi\le 600 \tag{5.5}$$

2D $f(x,y)=\dfrac{x^2+y^2}{4000}-\cos(x)\cos(\dfrac{y}{\sqrt{2}})+1$

Global minimum (x,y)=(0,0) $f=0$;

The standard PSO and improved PSO with different parameters are listed in Table 5.3. For better comparison of the rate of convergence with other algorithm, the GAs (Genetic Algorithms) is also listed.

The number of particles and generation are set to be 200 and 100 respectively. The corresponding GAs setting is that population and the cycle number are 200 and 100. Every algorithm ran 10 times and selected the best solution of the 10 times. For standard PSO the coefficients are fixed, the adjusting convergence depends on the Vmax, and here Vmax is set as 2% of range.

Table 5.3 Parameters used for standard PSO, improved PSO and GA in comparison of the rate of convergence

Parameters Name	Standard PSO	Improved PSO	Parameters Name	Genetic Algorithms
Constriction function χ	N/A	0.72	Crossover rate	1
Initial inertia weight	1 (fixed)	1	Mutation rate	1/n
Final inertia weight	1 (fixed)	0.4	SBX	20
Cognitive parameter $c1$	2	2	Polynomial Mutation	100
Social parameter $c2$	2	2	Selection Method	Tournament selection

Typical convergence of average fitness values as a function of generations is shown in Figure 5.2 The rate of convergence of standard PSO is better than GA. The improved PSO is the best among three algorithms. In the Figure (c) and (d), the improved PSO is still the best one but GA performed better than standard PSO. The reason is that the standard PSO has to adjust itself via Vmax, which is difficult to control.

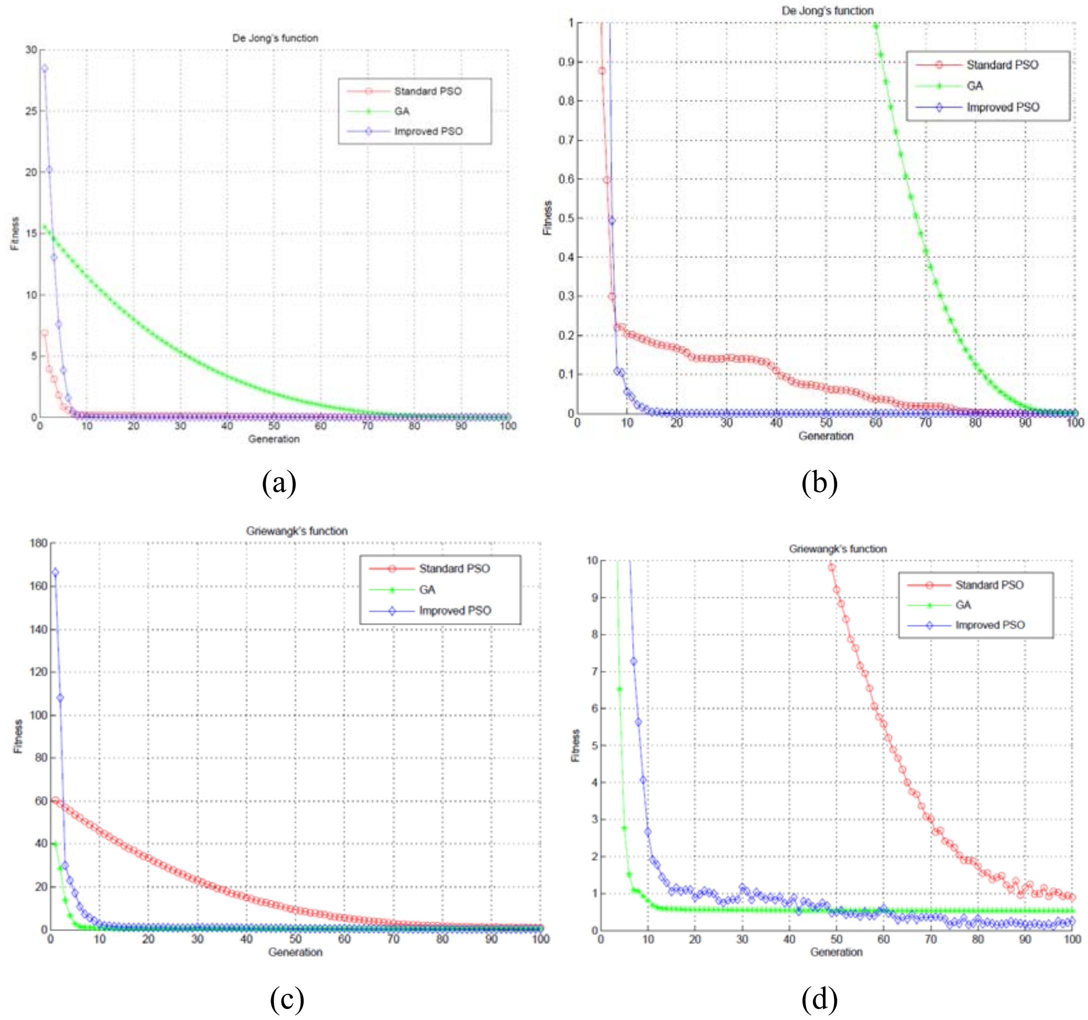


Figure 5. 2 Comparison of results for the rate of convergence of standard PSO, improved PSO and GA (a) De Jong's function; (b) De Jong's function with average fitness range from 0 to 1; (c) Griewangk's function; (d) Griewangk's function with average fitness range from 0 to 10;

Clerc and Kennedy (Clerc and Kennedy 2002) has given a mathematics analysis based on algorithm parameters and derived a reasonable set of parameters. Based on this, Trelea (Trelea 2003) provides a direct and simple guideline for parameter selection. A convergence triangle is given as a diagram (Figure 5.3). Trela transfers classic PSO formula to the following and makes c=d=1:

$$\begin{aligned} v_{k+1} &= a v_k + b_1 r_1 (p_1 - x_k) + b_2 r_2 (p_2 - x_k); \\ x_{k+1} &= c x_k + d v_{k+1}; \end{aligned} \qquad (5.6)$$

$$b = \frac{b_1 + b_2}{2};$$

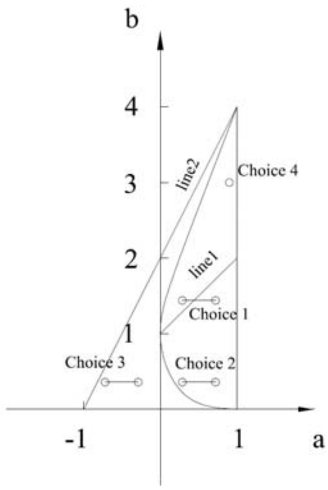
Corresponding to equation (2.1) in subsection **2.5.2**, *a* and *b* can be expressed as:

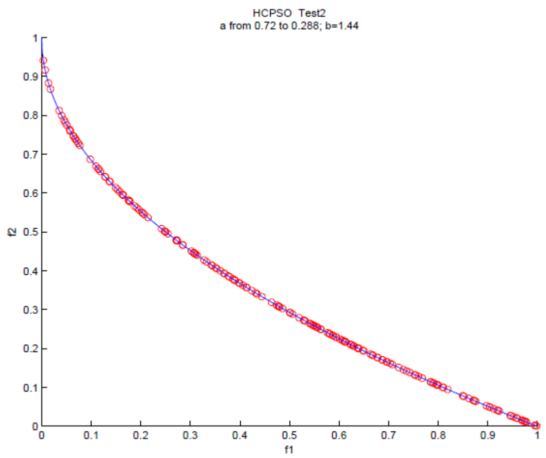
$$\begin{cases} a = \chi\omega; \\ b = \dfrac{\chi(c_1 + c_2)}{2} \end{cases} \tag{5.7}$$

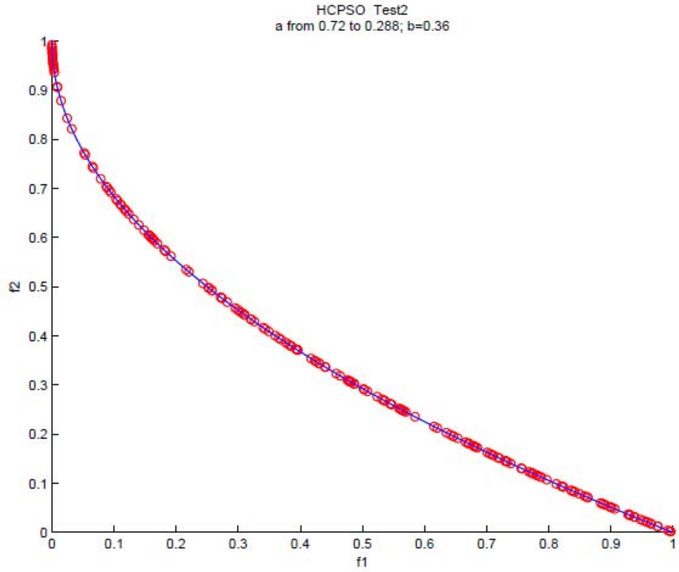
These parameters are critically important for PSO. The different parameters setting is given in Table 5.4. Test 2 is selected as a standard test function to evaluate the effort of parameters in HCPSO. In order to compare the final results (Figure 5.3), the number of particles and generation are set to 200 and 50 respectively.

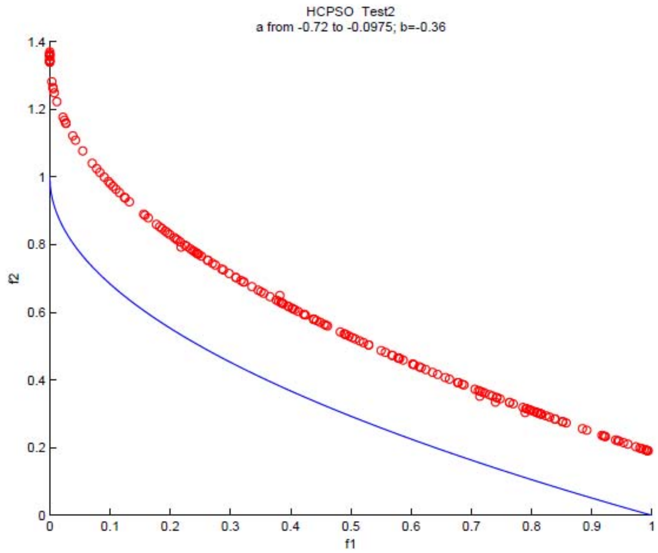
Table 5.4 Parameters used for HPCSO in comparison of the rate of convergence on multi-objective optimisation problem --- Test 2 in Table 5.1;

Parameters Name	Choice 1	Choice 2	Choice 3	Choice 4
Constriction Function χ	0.72	0.72	-0.72	1.5
Initial inertia weight	1	1	1	0.6
Final inertia weight	0.4	0.4	0.4	0.6
Cognitive parameter $c1$	2	0.5	0.5	2
Social parameter $c2$	2	0.5	0.5	2
a1	0.72	0.72	-0.72	0.9
a2	0.288	0.288	-0.288	0.9
b	1.44	0.36	-0.36	3






HCPSO Test2
a from 0.72 to 0.288; b=0.36
f2
f1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


HCPSO Test2
a from -0.72 to -0.0975; b=-0.36
f2
f1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
0.1
0.3
0.5
0.7
0.9

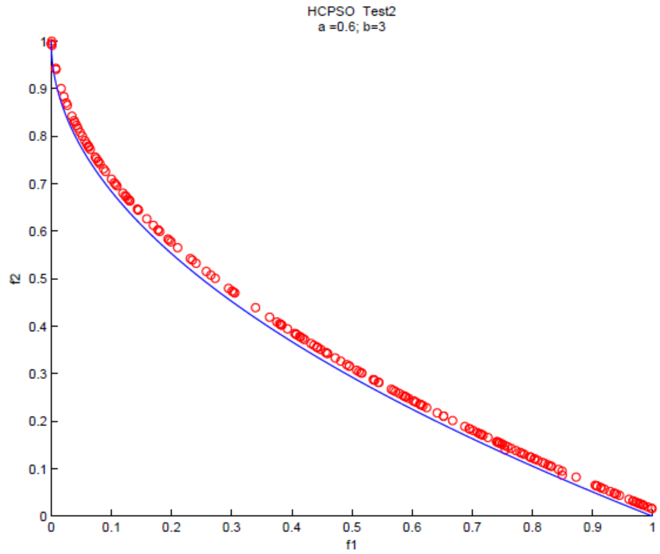


Figure 5. 3 Comparison of results for the rate of convergence of convergence on multi-objective optimisation problem --- Test 2 in Table 5.1;

The line 1 in Figure 5.3 is a zigzagging line and line 2 is a harmonic oscillation line. As can be seen from Figure 5.3, choices 1 and 2 fit well after optimisation, but choice 3 and choice 4 can not fit Pareto solutions. Figure 5.3 displays that the principle of Trelea is applicable to HCPSO.

5.4.3 Performance Metrics

Two evaluation criteria are selected:

(1) GD (Generational Distance) finds an average of the solutions of Q from P* (Q is solution and P* is a known set of the Pareto-optimal)

$$GD = \frac{(\sum_{i=1}^{|Q|} d_i^p)^{1/p}}{|Q|}$$

For a two objective problem (p=2), d_i is the Euclidean distance between the solution $i \in Q$ and nearest member of P*.

(2) Spread

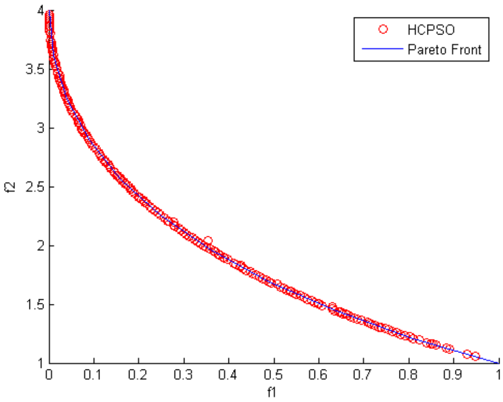
Deb et al. (Deb 2001) suggested the following metrics to alleviate the above difficulty

$$\Delta = \frac{\sum_{m=1}^{M} d_m^e + \sum_{i=1}^{|Q|} | d_i - \overline{d} |}{\sum_{m=1}^{M} d_m^e + | Q | \overline{d}}$$

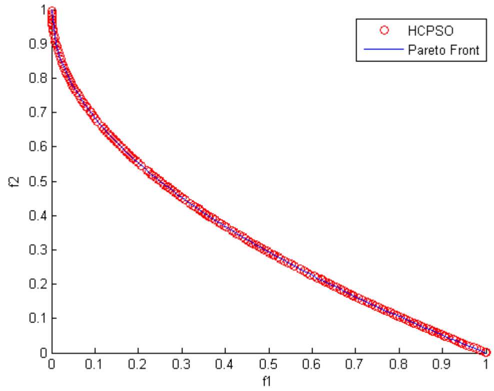
where $\overline{d}$ can be any distance measured between neighbouring solution and $\overline{d}$ is the mean value of the above distance measure. The parameter d_m^e is the distance between the extreme solutions of P* and Q corresponding to m[th] objective function.

5.4.3 Results

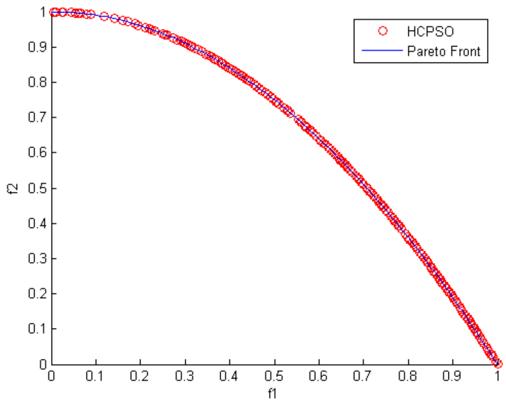
The HCPSO is run 10 times and real-coded NSGAII, a multi-objective genetic algorithm, is used for comparison. In HCPSO, the parameters are set as following: the population of swarm is 200 and generation is 100 iteration steps. c_1 and c_2 are set to 0.5 while ω is gradually decreased from 1.0 to 0.4. Vmax is set to the boundaries of decision variable ranges. χ is 0.72. The ε-disturbance has 3 steps. The number of sub-swarm is 2 and population of sub-swarm is 100. In NSGA II, the number of individuals is 200 and the number of generation is set to 100. The SBX (Simulated binary crossover) is used with $\eta_c = 10$ and the polynomial mutation is used with $\eta_m = 20$. The crossover and mutation probabilities are set to 0.9 and $1/n$ respectively. The parameters of NSGAII are selected according to prior study. The results of performance metrics of Test 1 to 6 are averaged and summarised in Table 5.5. The non-dominated solutions found by HCPSO for all test functions are displayed in Figure 5.4 and they all perform very well for both 2D and 3D objectives. The HCPSO gives last generation results as non-dominated solutions.



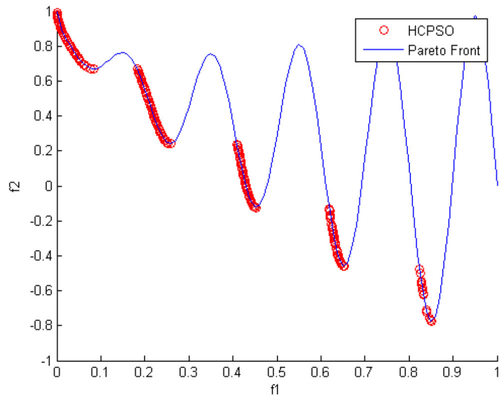
Test 1



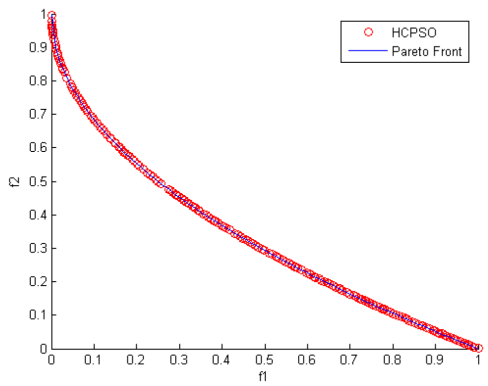
Test 2



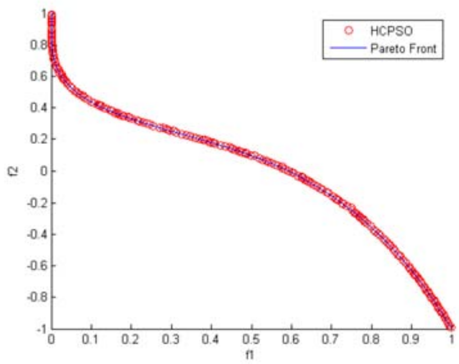
Test 3



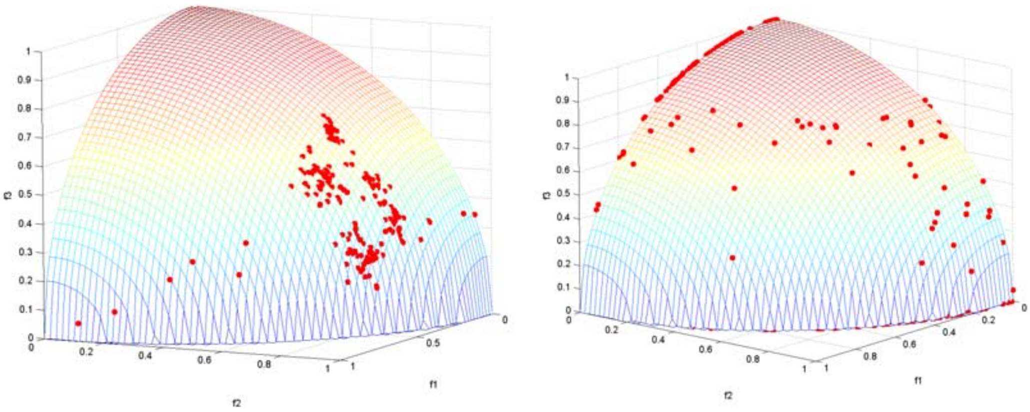
Test 4



Test 5



Test 6



Test 7 Test 8

Figure 5.4 Non-dominated solutions found by HCPSO on test functions in Table 5.1 and 5.2 with Pareto Fronts

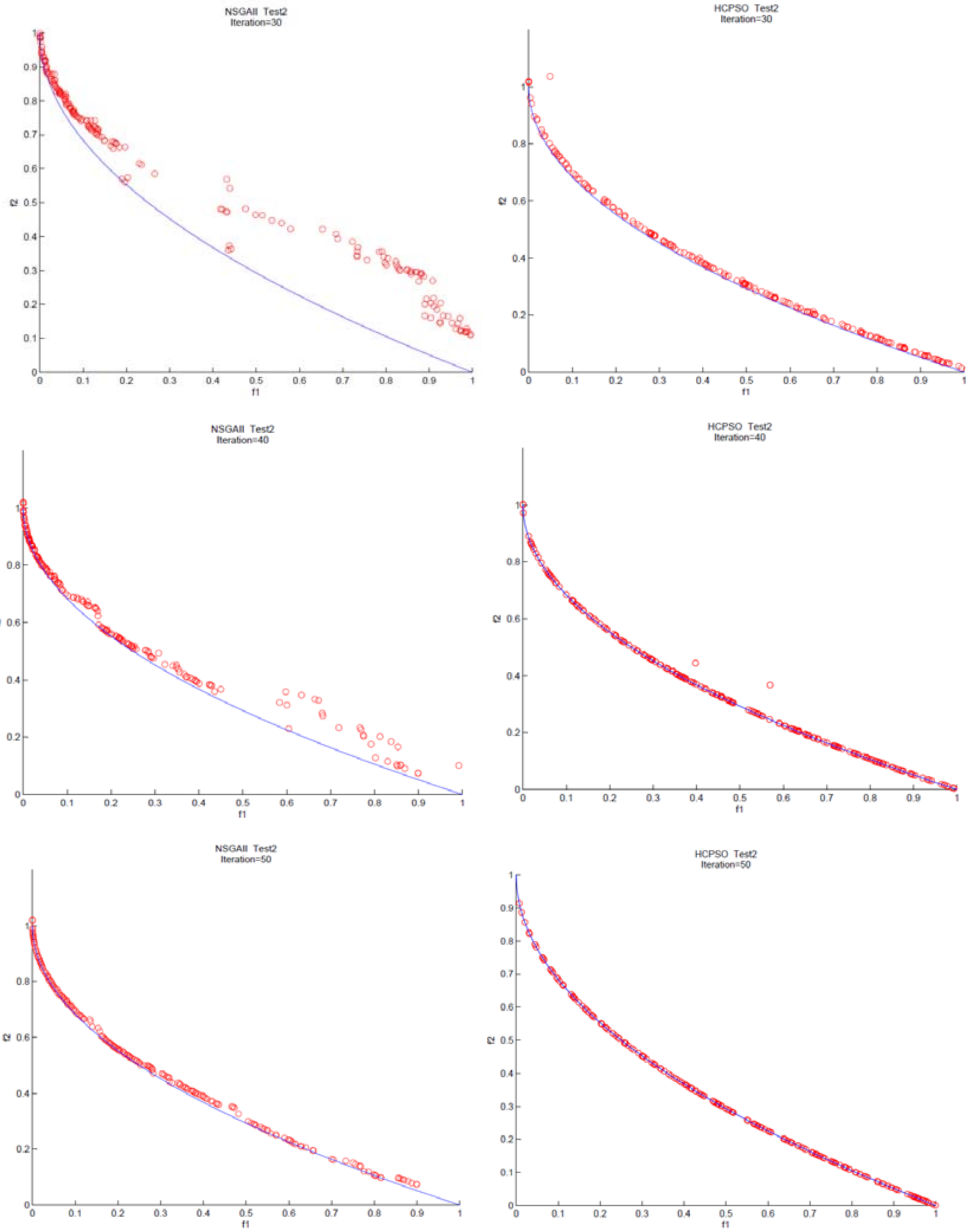
Table 5.5 Comparison of Mean and variance values of convergence metric GD and diversity metric Δ on six two-objective problems;

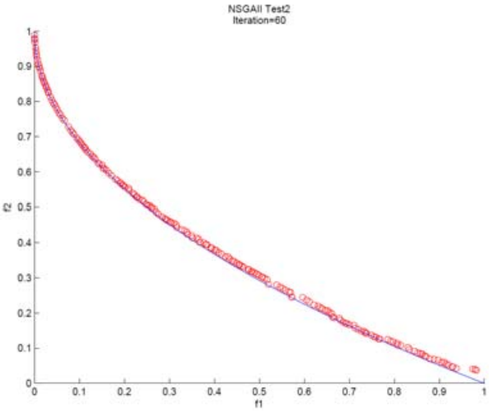
Algorithm	Test 1		Test 2		Test 3	
	GD	$\delta_{GD}{}^2$	GD	$\delta_{GD}{}^2$	GD	$\delta_{GD}{}^2$
HCPSO	2.35E-03	8.92E-05	1.12E-03	7.68E-05	8.22E-04	8.46E-04
NSGAII	2.42E-03	9.11E-05	1.85E-03	9.25E-05	9.57E-04	8.40E-04
	Δ	$\delta_{\Delta}{}^2$	Δ	$\delta_{\Delta}{}^2$	Δ	$\delta_{\Delta}{}^2$
HCPSO	4.12E-01	3.56E-02	3.86E-01	1.82E-02	3.82E-01	2.56E-02
NSGAII	4.29E-01	3.80E-02	4.72E-01	4.60E-02	4.80E-01	3.15E-02

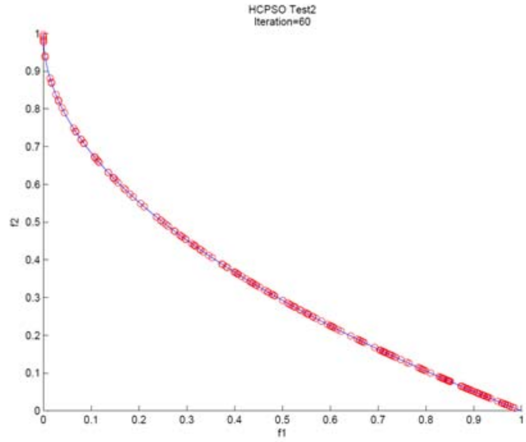
Algorithm	Test 4		Test 5		Test 6	
	GD	$\delta_{GD}{}^2$	GD	$\delta_{GD}{}^2$	GD	$\delta_{GD}{}^2$
HCPSO	6.19E-03	3.81E-03	6.72E-03	6.92E-03	7.13E-03	8.13E-03
NSGAII	2.91E-02	8.52E-02	3.60E-02	3.82E-02	6.20E-02	4.56E-02
	Δ	$\delta_{\Delta}{}^2$	Δ	$\delta_{\Delta}{}^2$	Δ	$\delta_{\Delta}{}^2$
HCPSO	6.12E-01	2.82E-02	6.88E-01	2.12E-02	7.47E-01	4.67E-02
NSGAII	9.80E-01	3.93E-02	7.41E-01	2.60E-02	8.32E-01	3.55E-02

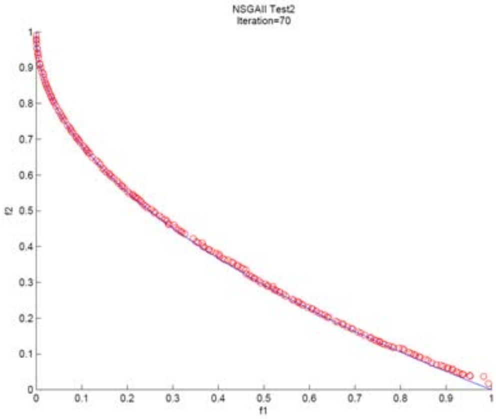
As can be seen in Table 5.5，HCPSO can perform very well for all six standard multi-objective optimisation test functions, which are tested in this chapter, in comparison to NSGAII.

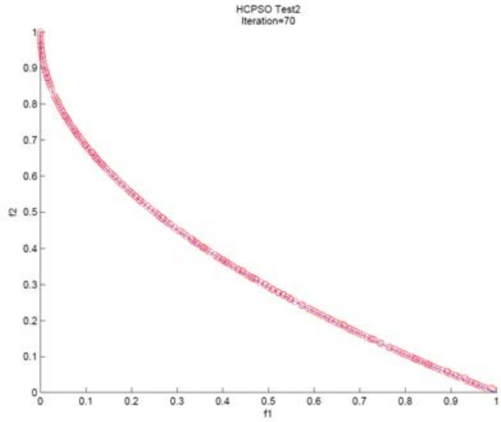
For checking the rate of convergence, the convergence history of two algorithms on test 2 is given in Figure 5.5. As it can be seen in Figure 5.5, the HCPSO is converging faster (approximately 60%) than NSGAII and this means a significant reduction in computing time.

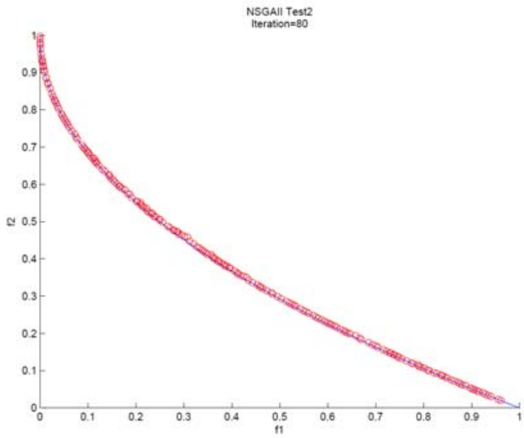


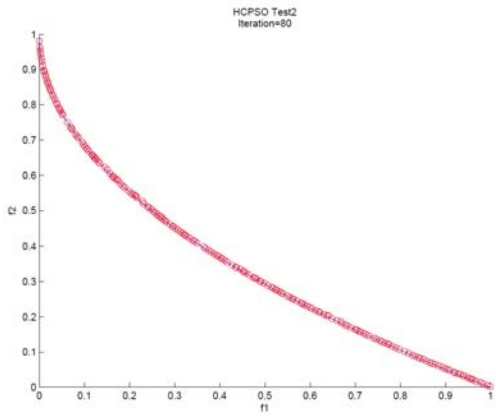

NSGAII Test2
Iteration=60
f2
f1


HCPSO Test2
Iteration=60
f2
f1


NSGAII Test2
Iteration=70
f2
f1


HCPSO Test2
Iteration=70
f2
f1


NSGAII Test2
Iteration=80
f2
f1


HCPSO Test2
Iteration=80
f2
f1

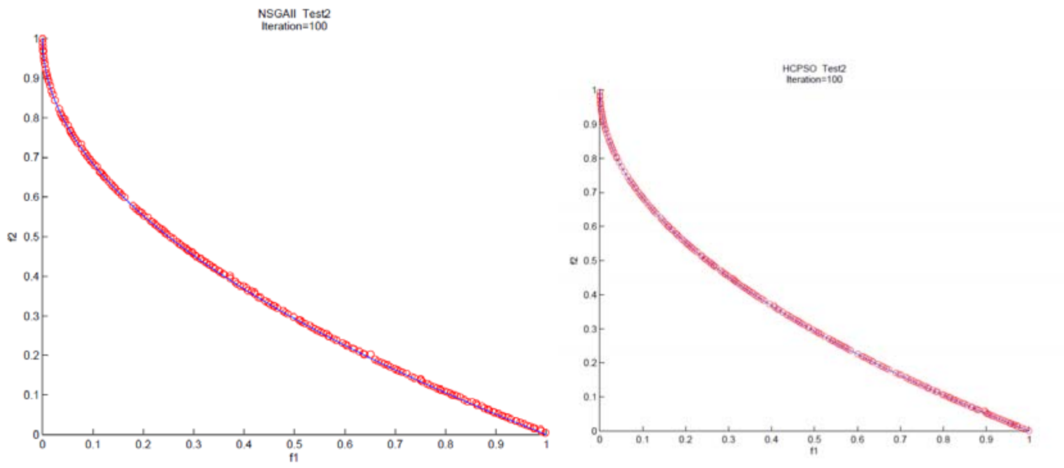


Figure 5.5 Comparison of convergence history of HCPSO and NSGAII on two-objective optimisation problem --- Test 2 in Table 5.1;

5.5 Conclusions

This chapter presents a new hybrid approach, HCPSO, which is a co-evolutionary based Multi-Objective Particle Swarm Optimisation methodology. The theory and workflow of algorithm are given together with the parameter selecting principle. The proposed algorithm is tested and validated successfully utilising the most widely used 2D and 3D test functions. In test functions, HCPSO converges significantly faster than NSGAII and provides 3D solutions using PSO comfortably.

This algorithm has two outstanding features for the proposed system. One is the sample parameter setting. This feature will help the system to easily control the optimisation process. The other is the internal learning ability. The optimisation itself has the ability of learning.

Chapter 6

Real-time Learning in Ship Design Environment

6.1 Introduction

Ship design is a complex and distributed optimisation process. The design conditions and the variables often change dynamically within both continuous and discrete ranges. A good and effective design support system should be able to observe these changes in time and can take actions to adjust accordingly. At the same time, the time of optimisation in ship design, which often runs for several weeks, is a critical problem for designers. So the design support system should reduce the design duration by speeding up the convergence intelligently. The prior experience provides a good way to solve both these problems. The support system can guide the direction of design and avoid the fault areas in advance via experience. The experience also can assist the system to adjust the uncertain variation of ship design environment.

In this chapter, the method, which simulates the human learning process, is introduced into ship design practice. Both the theory and application are explained within the context of the ship design process. Reinforcement learning as an important machine learning method is introduced here to solve the problem due to its good real time learning performance. The Q-learning as an idiographic approach of reinforcement learning is employed and embedded in ship design support system to realize and improve learning ability of the proposed system. Applications are presented in both the box model using manual calculation and the real ship model using the developed computer calculation. The advantages and disadvantages of the application of Q-learning are also discussed.

6.2 Aim of Real-time Learning in Ship Design

Learning in ship design is a new concept. As reviewed in Chapter 2, most of AI application in ship design focused on Case Based Reasoning. The machine learning merely has few applications and still is limited due to the programming problems. In this research, the machine learning will be introduced to the whole ship design process. In this chapter, the machine learning will be employed to solve the real time learning problem in optimisation support. This chapter is arranged as following:

(a) Analysis of the characteristics of ship design environment and process;
(b) The method of learning in ship design practice;
(c) The theory of reinforcement learning and Q learning with the adjustment for ship design;
(d) The realization of manual calculation for simple case;
(e) The realization of computer calculation for complex case;

6.3 Background and development

6.3.1 The development of ship design and design optimisation

The design optimisation is a basic and effective tool for decision based ship design. In early ship design period, the designers and engineers of different subjects worked together to develop an acceptable design for the design requirements and the design decision was made by experienced designers, who mainly employed their prior experience as their design criterions. They began to use some basic and simple optimisation approach to assist design action but it was not the main instrument at that time because there was a very simple method with limited objectives. So the optimisation is not powerful enough to support the design process.

On one hand, in preliminary design stage, every design condition would vary for different reasons, so the initial design may be changed many times for every condition changing. Designers had to collect data again and again to deal with this situation, which makes the design process very long and wastes a lot of time during the design work. On the other hand, because of the uncertainty and limitations of the experience, the final design could always not reach at the best level. The problem of improving design ability in this period focused on increasing efficiency.

With the wide application of computers in ship design process, more and more work is processed by the computer system. In optimisation research area, the optimisation technology has a great development and as a result the number of objectives has been increased in optimisation process. The optimisation subject has been recently used more widely in areas of ship design and also began to provide more support for decision making. But in general application, most optimisations still play process support as an extrinsic tool, which is independent of ship design process and usually is realized by the third party software.

The situation that the optimisation was not embedded in ship design support system but used as an extra tool which would cause two problems. The first one is that the optimisation may not match the requirements of ship design work and it needs designers to adjust input values, coefficients and criterions etc, that increases the degree of difficulty for ship designer to bring higher professional knowledge into optimisation. Because of the complex environment in ship design, it is very hard to modify an optimisation approach to adapt to every practical ship design. The other problem is the difficulty of selecting an appropriate optimisation approach. There are many approaches in the optimisation area but none of them is an ideal approach for all the problems. So for the best effect, the support system has to select the most appropriate optimisation approach, but every approach needs to adjust different parameters for different problems. Therefore how to select a good approach and give correct parameters setting become very important.

So the current problem of the optimisation application in ship design support system includes two aspects. First aspect is improving the efficiency and the other is reducing the time. For increasing the efficiency, two methods are employed in this study. One is forecasting the searching direction presented in this chapter and other is self-adaptive selecting optimisation approach presented in next chapter. The time is very important in ship design support system and too long running time would adversely affect the support ability of optimisation. For example, in a MOGA application of ship subdivision arrangement case, it may take couple of weeks to finish a full run. If design variables changed, the optimisation has to be done all over again. In this chapter, the machine learning approach is utilised to make a faster convergence speed.

In summary, there are two main problems when the optimisation is applied to the ship design support system. First one is the easy control of optimisation performance without changing optimisation algorithm itself, and the second one is the time. A decision support system is required to reduce the time to an acceptable level. Machine learning method can help decision support system on both aspects.

6.3.2 The characteristic of ship design optimisation

When applying machine learning to the ship design optimisation, the characteristics of ship design optimisation should be studied for selecting the most appropriate approach.

The first characteristic is that most variables in ship design process are discrete rather than continuous, which gives an advantage to control the optimisation. For the discrete variables, in relatively small area, the number of all choices of the design variables is finite. So in this situation, reviewing a fixed percentage optimal and known solutions and giving a forecast to the direction of other unknown solutions is possible. Discrete variables give a good basement for the application of machine learning, which is easier to realize than continuous variables in both theory and application.

The second characteristic is that the design environment is complex due to the frequent variable changing. Ship design is a systematic process, which includes hull, propeller, machinery, electrics, outfitting etc. And they are all fluctuating factors. The variation of one factor will result in the changes of other factors, which will directly lead to whole system re-optimisation. So the optimisation method should have excellent adaptability to the environment, which means that changing factors will not result in whole system re-optimisation but self-adjustment. In other word, the optimisation method should become aware of the changes of environment and can adjust automatically to these changes.

The last characteristic of ship design is the time cost which is the most important factor in decision based ship design process. Usually, the fitness evaluation of optimisation action in ship design optimisation is performed by another software. For example, in subdivision problem, the stability performance is always simulated in NAPA. Then the fitness function uses the stability performance simulation results as the fitness function to make the optimisation. In the next circulation, NAPA software will be called again. In this case, the most time costing process of design optimisation is caused by NAPA. So if the designers want to reduce the running time, the research emphasis should be put on reducing the times to call NAPA for each run.

In summary, the learning method should aim at these characteristics and provide an appropriate but simple and effective method to realize experience-sharing and give correct guidance.

6.4 Approach adopted in this chapter

In this method, the ship design process is treated as a repeating decision process. This method can not only be employed for a single ship design project but also can be utilised in the whole lifecycle design. The method improves the efficiency via sustainable development of the whole design process.

In a single run, the ship design optimisation process is divided into three parts according to memory theory. Every part makes a simulation of relevant function as shown in Figure 6.1.

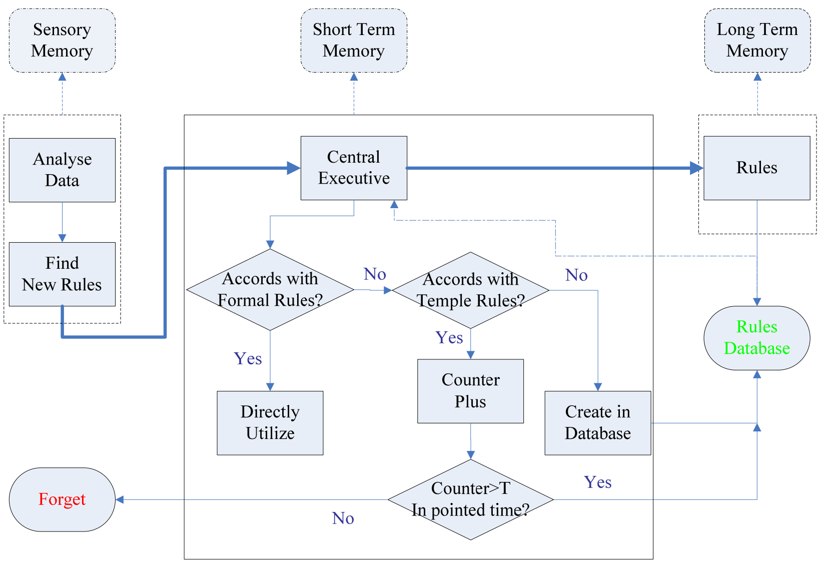


Figure 6. 1 The machine learning based ship optimal design in single run

Sensory memory part, as defined by psychologists as immediate memory, is used here to look for new rules in optimisation process. These new rules are established by trial and error. This is very helpful for design, because every time, the design task is different and the experience gained in one run maybe not be right for other designs. So first of all, the method should analyse the data and distinguish what type of design task it belongs to. Then the method should attend to derive more rules from data. These rules can be either selected to long-term memory and also may be abandoned to be forgotten.

Short-term memory, using the working memory theory, is the most important part in this method. This part is managed via a "central executive" centre, which firstly checks the new rule in the database to find whether it has already been the formal rule. The formal rules here mean the rules from mature knowledge, classification society, IMO, etc. The formal rules also include the rules which are found in previous designs and have been proven as reasonable and available resource. If the new rule is the formal rule, it can be directly used. If not, the central executive will continue to check whether the new rule belongs to temple rules. The temple rules mean the rules which have been proven correct at least twice. Every temple rule has a counter. If the new rule belongs to the temple rules, the counter of this temple rule will increase by 1. If not, the method will create a new rule and allot the relative counter. The central executive also checks the counter of every temple rule after a pointed time. If the counter exceeds the given numbers, the method will transfer the temple rules to formal rules. If not, the method will delete it.

Long-term memory, presented by Baddeley (Baddeley 1986), is stored sufficiently long duration in order to be accessible over a period anything more than a few seconds. In Figure 6.1, the "long-term memory" means that the method will store long term relationships discovered by the previous practice. This method can also be extended to whole design agency.

Reinforcement learning is one of the most important and classic machine learning approaches which solve the problem faced by an agent that must learn the behavior through trial-and-error interactions in a dynamic environment. "The work described here has a strong family resemblance to eponymous work in psychology, but differs considerably in the details and in the use of the word “reinforcement” (Kaelbling and Littman 1996). The reinforcement learning can be handled well with the real time learning and is easy to control without too many parameters. So it is introduced here for realizing the real time learning in ship design support system.

6.4.1 Reinforcement learning

In a classic reinforcement learning model, an agent is connected to the environment via the perception and action. In the model shown in Figure 6.2, B is an agent and T is the environment.

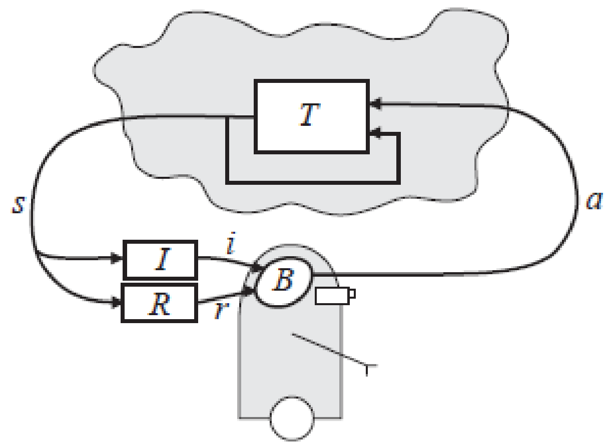


Figure 6. 2 The standard reinforcement learning Model (taken from Kaelbling etc, 1996)

In the first step, agent B receives an input ‘I’; in the second step, the agent B chooses an action ‘a’ to generate an output. This action ‘a’ changes the environment T and in third step, the value of this state transition is communicated to the agent B through a scalar reinforcement signal, ‘r’. The agent's behavior, B, should choose actions that tend to increase the long-run sum of the values of reinforcement signal. It can learn to do this over time by systematic trial and error, guided by a wide variety of algorithms.

In this model, there are two main factors which should be studied when it is applied in practice. The first one is what kind of evaluation the environment gives to the agent. The evaluation of the environment directly affects the direction of agent action. How to define the environment? How to evaluate the input signal and what kind of feedback should be given? These are all important questions. The second one is how to define reward and punishment. The strength and stability of reward and punishment signals are critical factors for the impression of learning method. When reinforcement learning is applied to the ship design support system, both of these factors must be considered carefully.

The reinforcement learning changes the agent's action via reward and punishment environment by assessing two key points: the environment evaluation and the environment reward strategy.

6.4.2 The Reinforcement learning in ship design decision support system

For the application in ship design decision support system, the function of a learning method is not constructing a new algorithm but a general method, which is independent of the optimisation algorithm. The function of learning focuses on two aspects. The first one is alarm mechanism; the learning method is able to make system not to touch or move far away from the forbidden area, for example, boundary limitation, etc. The second one is the guidance function; this means the learning method should give the direction of good solutions and avoid bad solution area as far as possible. Besides these two points, the learning method should also be able to solve the multi-agent communication in a simple way and to avoid introducing other additional factors.

There are three main reasons for employing the reinforcement learning approach to the ship design support system. The first one is the excellent real-time control function of reinforcement learning. In the optimisation process, the experience, rules and regulations are all fixed. If the designers want to change the performance of optimisation, a message should be given and the reinforcement learning can easily be realized. The second reason is the simplicity to operate. The reinforcement learning controls the direction of situation via rewards which means the designers can just modify the rewards to control the optimisation process. The last one is that the reinforcement learning can be independent from other optimisation approaches. The designers do not need to change the optimisation itself when applying the reinforcement learning.

The structure of environment is considered in the first place. There are three general principles. The first one is that the environment should be in modular form. This means that the environment should be constructed by different modules and these modules should have a uniform structure and can solve different problems via various combinations. The second one is that the environment should be simple and clear. The environment should use a simple and clear method with the distinct practical concept that makes this environment suitable for a decision system. The designers should be able to see clearly why the results are achieved. The last one is that the information sharing should be open to all the agents and should avoid the dialog between single pair agents. Although the communication and negotiation mechanism is a hot research area, for this research, the aim of the environment is the application in engineering. So the simple communication is more suitable. According to above principles, a model is proposed in Figure 6.3.

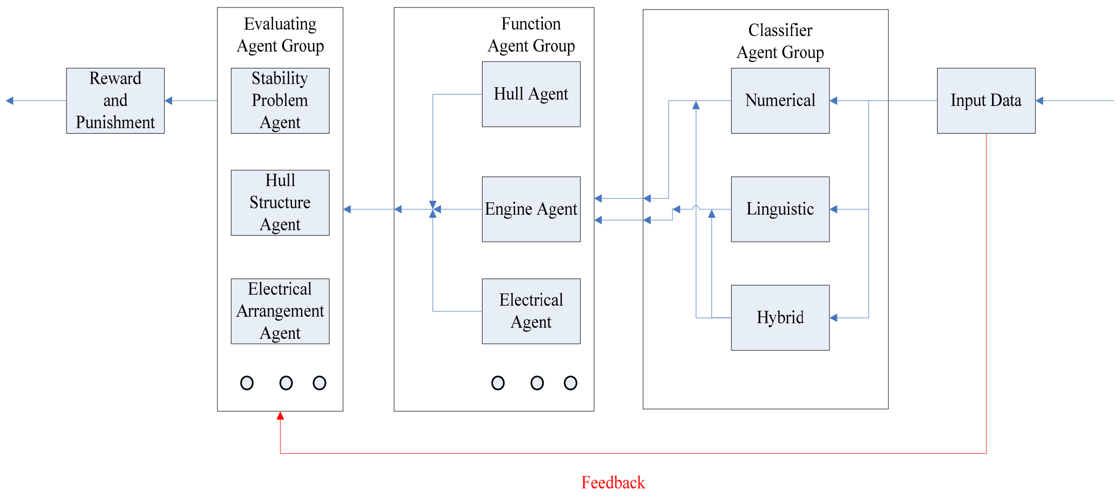


Figure 6. 3 Proposed reinforcement learning model for ship design decision support system

In this model, the action agent will give the integrated input information to the environment. Then, in the second step the classifier agent group would check this information and separate the data into two types: numerical and linguistic. The reason of this step is that the operational methods of these two types are totally different. Actually, most data are the numerical type. The linguistic type is very difficult to deal with. In the third step, two data types are used as input into Function Agent Group. In this group, the data would be classified as subjects like Hull, Electrical etc. The aim of this step is to provide organized information flow. Because the information sharing in this system is open and simple, the data must be in order and dimensionless for easy identification. Then the evaluating agent group would read corresponding data and provide the functions of reward and punishment. For better learning ability, a feedback to improve evaluating agent group is proposed in Figure 6.3 as the red line.

As the final step, the information sharing mechanism is considered. The principles used here are “open” and “simple”. The principle “open” means that all information is open to the members in the system and the “simple” means that the information sharing mechanism should be easy to realize and should try not to introduce new factors. There are different agent groups and sub agents with different functions just like a society. So the learning among ship design support system becomes a social learning. Because the theory of social learning is very complex and immature, for a decision support system, a simplified method is more suitable. A public board is set here just like a billboard to tell everything to everyone. All agents read the information from this public board and also write the results on it, which means that there is no communication between any two agents as all information is on the board.

This model has a number of advantages as explained below. Firstly, all the information is open to anyone including the agents in the environment and designers, who can be seen as external agents. This helps information maintenance and modification, which means that all the information exchange can be watched by designers and also can be adjusted manually. Secondly, non-communication between agents will reduce the complexity of the system. The communication among agents of the social learning is very complex. So here, all the information is given to public board. The agent just needs to spend time to test the publishing board, which is a single direction. It just needs to contact public board rather than contacting other agents. Lastly, through the uniform format, the information becomes simple and easily recognized.

There are two disadvantages of this model. One is that the public board may turn out to be large and complex. The other is the agent would have to "ping" to public board which will make traffic problem and block the system. The "ping" here means the system send a test signal to test every agent.

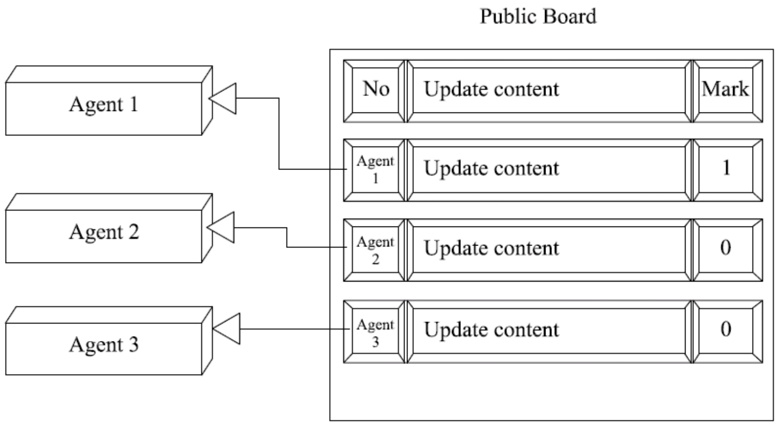


Figure 6. 4 Public board model of ship design decision support system

Figure 6.4 presents the model of public board. There are three main key aspects in this model. The first one is the agent must have a clear output format. For reducing the complexity and increasing stability of the system, the agent must be clear about to which agent, the output result should be sent to. The second one is that the public board design includes three parts: number, context and read mark. The information sent by agent firstly checks the number and finds its own location. Then check the read mark, if read mark is '*there is a message which has not been read*', which represented by '1', the information will wait until the read mark is clear, which represented by '0'. After the read mark is clear, the agent can read information to update. The third one is that the traffic of the system. Here, the sequence method is accepted, which means the system will read the public board one by one in despite of the clear read mark. In other words, the agent just read once and then move to the next. So the agent would not waste time to test the read state.

6.4.3 Q learning

6.4.3.1 Introduction of Q learning

There are many approaches in reinforcement learning area, but Q-learning, as an excellent approach, is selected to realize the real time learning in this study.

Q-learning (Watkins 1989), (Watkins and Dayan 1992) is one of the important reinforcement learning approaches. It works by learning an action-value function that gives the expected utility of taking a given action in a given state and following a fixed policy thereafter. An important advantage of Q-learning is that it is able to compare the expected utility of the available actions without requiring a model of the environment.

As a form of model-free reinforcement learning, the requirement of Q-learning is loose for the environment, but this does not mean Q-learning is applicable for any situation (Watkins and Dayan 1992). With the development of Q-learning theory, the research in continuous mathematical model has made a progress. But the discrete and finite mathematical environment still is the main topic of application for Q-learning. So a discrete environment with finite steps changing is a good foundation of the Q-learning, which just fits to the characteristic of ship design optimisation presented in Section 6.3.2.

The Q learning method has developed very quickly in recent years and has extensive application in engineering, business, management *etc*.

6.4.3.2 Analysis of Q learning

The theory of Q-learning is simple and clear but its mathematical proof is complex and involves many disciplines. In order to give a systematic and general understanding, a conceptual framework combined with mathematical discussion is presented in this section.

First of all, the method still begins from the concept of reinforcement learning. Reinforcement learning as an unsupervised learning method has a difficult point, which is the learning system that can not be taught whether the actions it performed are good or bad because there are no ‘teachers’. For example, in ship design optimisation, the aim of a learning system is to avoid the failure of stability via calculating the Index A with Transverse Bulkhead as variables. If the final result is failed, how to decide which modification Transverse Bulkheads should be responsible for the failure? A method in mathematical field called dynamical programming can solve this problem. The dynamic programming involves just two basic principles. “First, if an action causes something bad to happen immediately, then the system learns not to do that action in that situation again. The second principle is that if all the actions in a certain situation lead to bad results, then that situation should be avoided. With these two principles, the reinforcement learning also can do any number of tasks”. (Harmon and Harmon 1996)

The Q-learning is developed from the theory of dynamic programming. The dynamic programming itself has the strict mathematical proof. For better understanding Q-learning, the essence of dynamic programming is taken into account. For the dynamic programming, the primary objective of learning is to find the correct mapping from states to state values. In other words, the dynamic programming tries to find the relationship between the states and the expression values of state.

Let us assume that:

$V^*(x_t)$ is the optimal value function where x_t is the state vector;

$V(x_t)$ is the approximation of the value function;

γ is a discount factor in the range [0,1] that causes immediate reinforcement to have more importance (weighted more heavily) than future reinforcement;

$e(x_t)$ is the error in the approximation of the value of the state occupied at time t;

Without loss of generation, $V(x_t)$ will be initialized to random values without any information about $V^*(x_t)$. This means $V(x_t)$ is equal to the sum of $V^*(x_t)$ and $e(x_t)$. As expressed in Equation (6.1);

$$V(x_t) = e(x_t) + V^*(x_t); \qquad (6.1)$$

for time t+1, it is the same as Equation (6.2)

$$V(x_{t+1}) = e(x_{t+1}) + V^*(x_{t+1}); \qquad (6.2)$$

$V^*(x_t)$ is the sum of the reinforcements when starting from state x_t and performing optimal actions until a terminal state is reached. By this definition, a simple relationship exists between the values of successive states x_t and x_{t+1} as Equation (6.3). Here γ is used to exponentially decrease the weight of reinforcements received in the future.

$$V^*(x_t) = r(x_t) + \gamma V^*(x_{t+1}); \tag{6.3}$$

$V(x_t)$ has the same relationship as in Equation (6.4)

$$V(x_t) = r(x_t) + \gamma V(x_{t+1}); \tag{6.4}$$

Then by substituting Equations (6.1) and (6.2) into (6.4), one can get Equations (6.5) and (6.6).

$$e(x_t) + V^*(x_t) = r(x_t) + \gamma(e(x_{t+1}) + V^*(x_{t+1})); \tag{6.5}$$

$$e(x_t) + V^*(x_t) = r(x_t) + \gamma e(x_{t+1}) + \gamma V^*(x_{t+1}) \tag{6.6}$$

Substituting Equation (6.3) into (6.6), one can get Equation (6.7).

$$e(x_t) = \gamma e(x_{t+1}) \tag{6.7}$$

From Equation (6.7), if it is true for all x_t, then the approximation error in each state is required to be zero. So the process of learning is the process of finding a solution to Equation (6.4) for all states (which is also to Equation (6.7)).

Now, if it is assumed that the function approximator used to represent V^* is a lookup table, then one can find the optimal value function by performing sweeps through state space, updating the value of each state according to Equation (6.8).

$$\Delta w_t = \max_u(r(x_t, u) + \gamma V(x_{t+1})) - V(x_t); \tag{6.8}$$

in which u is the action performed in state x_t and causes a transition to state x_{t+1}.

$$e(x_t) = \max_u (r(x_t, u) + \gamma V(x_{t+1})) - V(x_t) \qquad (6.9)$$

So the aim of learning is to find $e(x_t) = 0$.

For Q-learning, a deterministic Markov Decision Process (MDP) is one in which the state transitions are deterministic. In a nondeterministic MDP, a probability distribution function defines a set of potential successor states for a given action in a given state. If the MDP is non-deterministic, the iteration of value requires that we find the action that returns the maximum expected value. Theoretically, value iteration is possible in the context of nondeterministic MDP. However, in practice, it is computationally impossible to find the necessary integrals without additional knowledge or some modification. Q-learning solves the problem by taking the maximum value over a set of integrals. Rather than finding a mapping from state to state values (as in value iteration), Q-learning finds a mapping from state/action pairs to values (called Q-values). Instead of having an associated value function, Q-learning makes use of the Q-function. In each state, there is a Q-value associated with each action. The definition of a Q-value is the sum of the (possibly discounted) reinforcements received when performing the associated action and then following the given policy thereafter. Likewise, the definition of an optimal Q-value is the sum of the reinforcements received when performing the associated action and then following the optimal policy thereafter.

Q-learning differs from value iteration in that it doesn't require that in a given state each action be performed and the expected values of the successor states be calculated.

$$Q(x_t, u_t) = r(x_t, u_t) + \gamma \max_{u_{t+1}} Q(x_{t+1}, u_{t+1}) \qquad (6.10)$$

So the learning process is seeking the solutions to Equation (6.10).

Considering Q-learning, before learning has started, Q returns a fixed value, chosen by the designer. Then, each time the agent is given a reward (the state has changed). New values are calculated for each combination of a state s from S, which is statement sets, and action a from A, which is action sets. The core of the algorithm is a simple value iteration update. It assumes the old value and makes a correction based on the new information as shown in Equation (6.11).

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \eta \times \left[r_t + \gamma \max Q(s_{t+1}, a) - Q(s_t, a_t) \right] \qquad (6.11)$$

where r_t is the reward given at time t, η ($0 < \eta \leq 1$) is the learning rate, may be the same value for all pairs. The discount factor γ is $0 \leq \gamma < 1$. Equation (6.11) is equivalent to:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t)(1-\eta) + \eta \left[r_t + \gamma \max Q(s_{t+1}, a_t) \right] \qquad (6.12)$$

6.4.3.3 The Application of Q learning in Ship Design Optimisation

The application of Q learning in this research requires two functions. The first function is Q learning, which will be used in an independent programming which means the Q learning can be applied into any ship design optimisation. The second function is that is the Q learning can adopt changes according to environment. The Q learning should adjust to the environment and change the reward or punishment. Figure 6.5 presents the detailed work flow of proposed Q-learning in ship design optimisation.

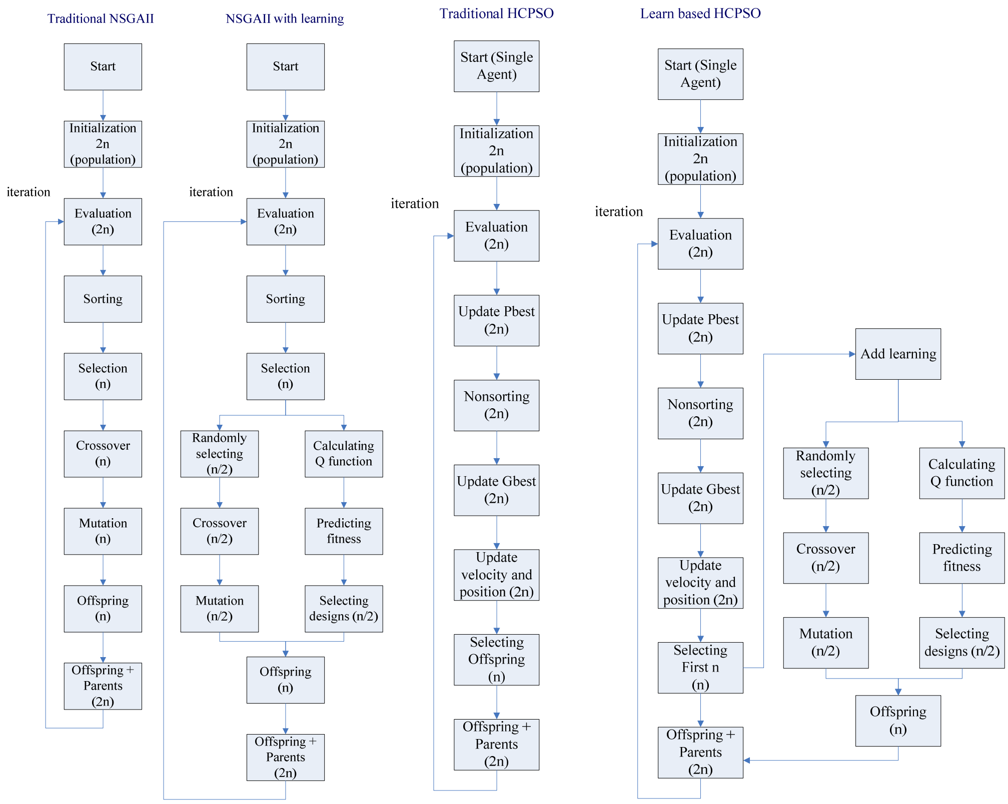


Figure 6. 5 The work flow of Q-learning in ship design optimisation

6.5 Application in Ship Design Environment

The ship design optimisation is a dynamic process and this process always has two limitations: run time and parameters setting. With regards to run time, an optimisation operation usually needs to link with the third party software to make the simulation and evaluation. This usually costs large time in calculation and simulation. For example, in ship subdivision problem it took a couple of weeks for three objectives optimisation in current mainstream configuration of a single processor. With regards to parameters setting aspect, the problem focuses on the limiting numbers of population and generation. The current approaches utilised in ship design optimisation are heuristic method such as GA approaches. For GAs, large population and large amount for repeated calculations are two important factors. But in application of ship design, these two factors have limits. The main reason of this problem is that the evaluation of fitness has to rely on the third software which would cost most of the time in current conditions of technology. The large population and large generation obviously elongate the running time of optimisation process, and also elongate the time of whole design work. In order to solve these two problems, Q-learning is applied here to reduce the time and obtain Pareto front in fewer numbers of steps because it can help correct optimisation direction and keep away from the low efficiency area.

For some brief propose, a simple barge subdivision optimisation problem is employed in this study to evaluate the calculation ability of Q learning. This barge ship can be seen as a box model and the whole case study will be divided into two parts: On is to check the ability of control single objective which will be proposed in 6.5.2 and other is to check the ability of control multi-variables which will be introduced in 6.5.3. Taking into account the complexity of calculation, the multi-objective problem will be tested in chapter 9 with real world problem.

The method accepted in this case study is calculation of the real value of total deck area. Then some of values are randomly selected as input data. The system will calculate predicted value of other deck areas via Q-learning. The predicted value will be compared with the real deck area to judge the ability of Q-learning.

6.5.1 Introduction of optimisation on box model

Model: A barge model with length 150m, breadth 40m and depth 10m. (Figure 6.6): In this model, the locations of Transverse Bulkhead (TB) including TB01, TB02 and TB03 can be changed when the locations of Longitudinal Bulkhead (Kaelbling and Littman) including LB01, LB02 and LB03 are also variable. The Longitudinal Bulkhead LB is fixed. The bilge area between two lines of longitudinal bulkhead decides the capacity of barge. When adding the height of barge, the corresponding space also represents the capacity. So in this example, the bilge area and space are selected as the objectives, which are seeking the maximum and the variables will be improved step by step.

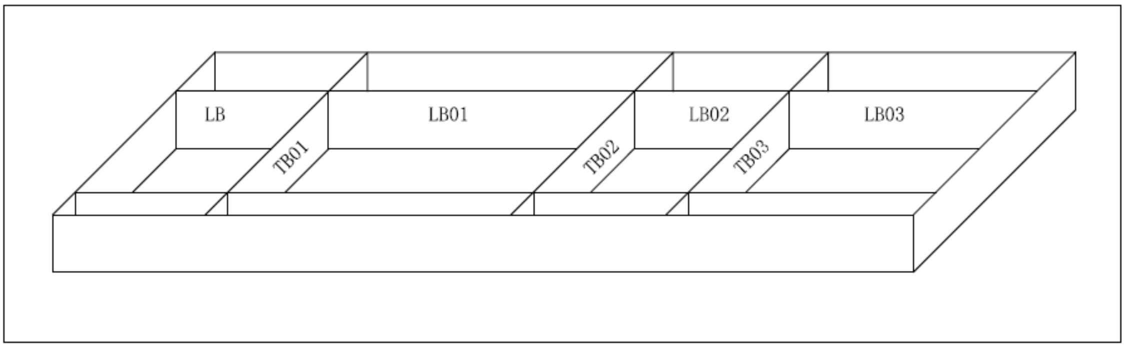


Figure 6. 6 The barge model for calculation

Variables: Transverse Bulkhead (TB) and Longitudinal Bulkhead (Kaelbling and Littman). Constraint: Table 6.1 shows the dimension constraints of this model.

As presented in Figure 6.7, the Longitudinal Bulkheads (Kaelbling and Littman) including LB01, LB02 and LB03 have the same boundaries, which vary between 0 m and 10 m, but every LB change independently. In Table 6.1, the step of variables is given as 5 m, which means the LB just has three choices: 0 m, 5 m and 10 m. As shown in Figure 6.7, the Transverse Bulkheads (TB) have different boundaries. TB01 has the boundaries from 35 m to 45 m and as given in Table 6.1, while the step of TB is 5 m. So the TB01 has three choices: 35 m, 40 m and 45 m. Correspondingly, TB02 has three choices: 75 m, 80m and 85m, when TB03 has three choices: 115 m, 120m and 125m.

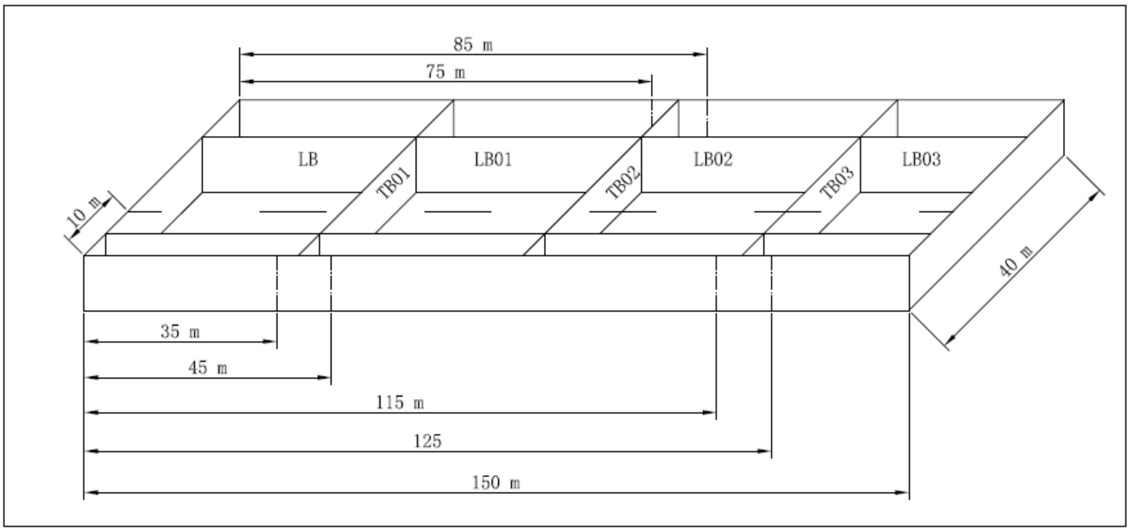


Figure 6. 7 The barge model constraints

Table 6. 1 The barge model constraints

No	Optimisation Variables	Limits		Step Size
		Lower (m)	Upper(m)	(m)
1	TB01	35	45	5
2	TB02	75	85	5
3	TB03	115	125	5
4	LB01	0	10	5
5	LB02	0	10	5
6	LB03	0	10	5

6.5.2 Case study of box model 1: single objective with two variables

Aim of case study of box model 1

This experiment includes one objective and two optimisation variables. The aim is to test the learning ability of Q-learning on single objective optimisation problem.

Optimisation Objective:

Maximum Deck Area

Optimisation Variables:

It is assumed that TB02, TB03 is fixed to 80 m, 120m; LB, LB02, LB03 is fixed to 0, 0 (Figure 6.8), so there are only two variables: TB01 and LB01.

Table 6. 2 The barge model optimisation variables

No	Optimisation Variables	Value (m)	Step (m)
1	TB01	35-45	5
2	LB01	0-10	5

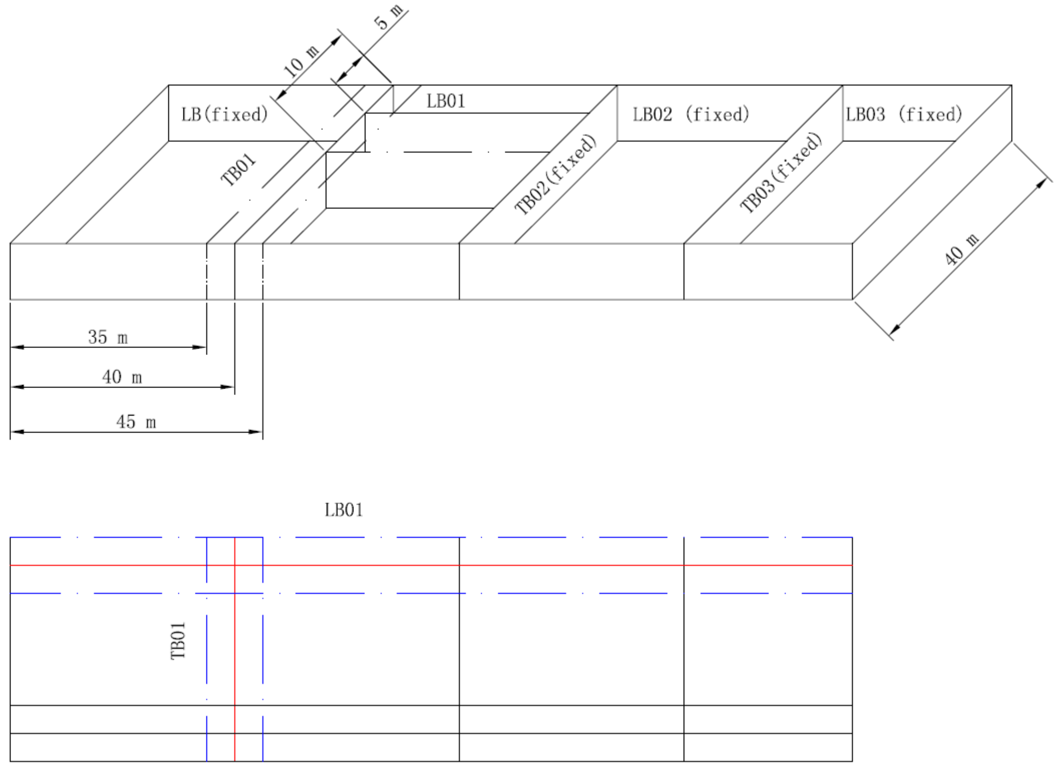


Figure 6. 8 The barge model of Q learning (single objective with two variables)

The variables and steps are shown in Table 6.2. The variable TB01 is limited from 35 m to 45m with the step of 5m, which means the TB01 has three choices to change: 35 m, 40 m and 45 m. For the variable LB01, the changing range is the same, from 35 m to 45 m and the step still is 5 m. The Figure 6.8 provides the visual aid for the variables and steps, the blue dash-dot line is the limitations of variables and red line is the central location of the variables.

Set parameters and environment reward matrix **R**

$S_{D\max}$ ---Max deck area;

$S_{D\max} = 150 \times 40 = 6000 m^2$

$S_{D\min}$ ---Min deck area;

$S_{D\min} = 35 \times 40 + (80 - 35) \times (40 - 10 \times 2) + 40 \times 70 = 5100 \ \ m^2$

$S_{D\text{stand}}$ ---Stand deck area, this value is defined by designer for the convenience of calculation according to $S_{D\max}$ and $S_{D\min}$. This value is used as the comparison standard for deck area calculation to make $S(\text{deck area})$ non-dimensionalise.

For example, in this case, $S_{D\text{stand}}$=5000 m^2 is selected, which is smaller than the minimum value of $S_{D\max}$ and $S_{D\min}$.

So reward matrix **R** can be obtained from S (deck area) $S(\text{deck area}) - S_{D\text{stand}}$;

Parameters
$\eta = 0.15$;
$\gamma = 0.95$;

Calculate the real value of deck area as shown in Table 6.3:

$$S_{AI} = 35\times40+(80-35)\times(40-0\times2)+40\times(150-80)=6000;$$
$$S_{AII} = 35\times40+(80-35)\times(40-5\times2)+40\times(150-80)=5550;$$
$$S_{AIII} = 35\times40+(80-35)\times(40-10\times2)+40\times(150-80)=5100;$$
…

Table 6. 3 The deck area of barge model Q learning, single objective with two variables

		A	B	C
		Transverse Bulkhead01	Transverse Bulkhead01	Transverse Bulkhead01
		35	40	45
I	Longitudinal Bulkhead01 0	6000	6000	6000
II	Longitudinal Bulkhead01 5	5550	5600	5650
III	Longitudinal Bulkhead01 10	5100	5200	5100

The values in Table 6.3 are the real deck area of different bulkhead location according to Table 6.2. Now, let us select three Rs randomly and calculate these three values of R according to

$$R = C_1\times(S_{\text{deck area}} - S_{\text{Dstand}})/S_{\text{Dstand}}; \qquad (6.13)$$

C_1 is the constant and here is 100. The function of C_1 is scaling up the original R for the calculation convenience. The aim of Equation 6.13 is to make nondimensionalisition. Too large or too small numerical value may cause the calculation error, so non-dimensionalisation is necessary for Q-learning.

The randomly selected values are AI, BII and CIII, so the R values are calculated as follows.

$R_{AI} = 100 \times (6000 - 5000) / 5000 = 20;$

$R_{BII} = 100 \times (5600 - 5000) / 5000 = 12;$

$R_{CIII} = 100 \times (5100 - 5000) / 5000 = 2;$

The detailed R value in this calculation is shown in Table 6.4.

Table 6. 4 The R value deck area of barge model Q learning, single objective with two variables;

		A	B	C
		Transverse Bulkhead01	Transverse Bulkhead01	Transverse Bulkhead01
		35	40	45
I	Longitudinal Bulkhead01 0	20		
II	Longitudinal Bulkhead01 5		12	
III	Longitudinal Bulkhead01 10			2

Assuming:

The optimisation variables have to move one step in order to change in every step. For example, the V_{AII} can move to V_{AI}, V_{AIII} and V_{BII} but can not move to V_{BI} or V_{BII}.

So

$$R = \begin{bmatrix} 20 & - & - \\ - & 12 & - \\ - & - & 2 \end{bmatrix} \text{ and } Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Step1

Randomly Select Q_{BII} $Q_{STEP1} = \begin{array}{c|ccc|} & A & B & C \\ I & Q_{AI} & Q_{BI} & Q_{CI} \\ II & Q_{AII} & Q_{BII} & Q_{CII} \\ III & Q_{AIII} & Q_{BIII} & Q_{CIII} \end{array}$

According to the assumption, in this time Q_{BII}=0, and next time, Q_{BII} can move to Q_{AII}, Q_{BI}, Q_{BIII}, Q_{CII}.

Obviously, in next time the $Q_{\max}$ =Max(Q_{AII}, Q_{BI}, Q_{BIII}, Q_{CII}) = 0.

According to equation (6.12),

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t)(1-\eta) + \eta\left[r_t + \gamma \max Q(s_{t+1}, a_{t+1})\right]$$

$$\begin{aligned} Q_{BII}(s_t, a_t) &= 0 \times (1-\eta) + \eta\left[r_t + \gamma \times 0\right] \\ &= \eta \times r_t \\ &= 0.15 \times 12 = 1.8 \end{aligned}$$

So Q value can be updated to

$$Q_{STEP1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.8 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Step2

This time, Q_{AII} is randomly selected. $Q_{STEP2} = \begin{array}{c|ccc|} & A & B & C \\ I & Q_{AI} & Q_{BI} & Q_{CI} \\ II & Q_{AII} & Q_{BII} & Q_{CII} \\ III & Q_{AIII} & Q_{BIII} & Q_{CIII} \end{array}$

According to the assumption, in this time Q_{AII}=0, and next time, Q_{AII} can move to Q_{AI}, Q_{BII} and Q_{AIII}.

When checking the Q_{STEP1}, it can be achieved that $Q_{\max}$ =Max(Q_{AI}, Q_{BII}, Q_{AIII}) = 1.5.

According to equation (6.12),

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t)(1-\eta) + \eta\left[r_t + \gamma \max Q(s_{t+1}, a_{t+1})\right]$$

$$\begin{aligned} Q_{BII}(s_t, a_t) &= 0 \times (1-\eta) + \eta\left[r_t + \gamma \times 1.8\right] \\ &= \eta\left[r_t + \gamma \times 1.8\right] \\ &= 0.15 \times [0 + 0.95 \times 1.8] \\ &= 0.2565 \end{aligned}$$

So Q value is updated to

$$Q_{STEP2} = \begin{bmatrix} 0 & 0 & 0 \\ 0.2565 & 1.8 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Then calculation is performed according to equation (6.12) and run can be repeated for 1000 times, the results are listed in Table 6.5;

Table 6. 5 The results of value deck area for barge model Q-learning, single objective with two variables

(a)

		A Transverse Bulkhead01 35	B Transverse Bulkhead01 40	C Transverse Bulkhead01 45
I	Longitudinal Bulkhead01 0		454	418
II	Longitudinal Bulkhead01 5	452		454
III	Longitudinal Bulkhead01 10	409	442	

(b)

		A	B	C
		Transverse Bulkhead01	Transverse Bulkhead01	Transverse Bulkhead01
		35	40	45
I	Longitudinal Bulkhead01 0		17.27%	15.90%
II	Longitudinal Bulkhead01 5	17.19%		17.27%
III	Longitudinal Bulkhead01 10	15.56%	16.81%	

The Table 6.3 indicates that BI, CI and AII, CII provide bigger deck areas. So the aim of this experiment is that the proposed algorithm should indicate the optimisation direction to BI, CI, AII and CII. In Table 6.5 (a), the black numbers in red area are the values obtained from calculations which are the probability calculated by Q-learning. These numbers are not the accurate values of the future in this point but are the moving directions of optimisation variables. Because after 1000 calculations, the values in Table 6.5 (a) are very big. For better understanding, Table 6.5 (a) is transferred to Table 6.5 (b) as percentage format via being divided by amount. Table 6.5 (b) provides the probability of moving to bigger deck area. The new points, which are created in optimisation, should spread more in red areas and the old points should move to red areas. For example, the CIII point should move to CII firstly because the value of CII is bigger than BIII. From Table 6.3, it can be seen that the real value of CII is better than BIII. This demonstrates that the Q learning algorithm can correctly forecast the direction to the field of the maximum deck area

In order to improve the forecast ability, the ability of changing the Transverse Bulkhead and Longitudinal Bulkhead at the same time is considered. The initial value and parameters setting are the same as above case including the R values. After 1000 runs, the results listed in Table 6.6

Table 6. 6 Improvement of the results I for deck area value of barge model; Q-learning, single objective with two variables;

(a)

		A Transverse Bulkhead01 35	B Transverse Bulkhead01 40	C Transverse Bulkhead01 45
I	Longitudinal Bulkhead01 0		504	516
II	Longitudinal Bulkhead01 5	508		518
III	Longitudinal Bulkhead01 10	507	502	

(b)

		A Transverse Bulkhead01 35	B Transverse Bulkhead01 40	C Transverse Bulkhead01 45
I	Longitudinal Bulkhead01 0		16.50%	16.89%
II	Longitudinal Bulkhead01 5	16.63%		16.96%
III	Longitudinal Bulkhead01 10	16.60%	16.43%	

The Table 6.6 is the same as Table 6.5 and from Table6.6, it can be clearly seen that the proposed system can provide the direction corrects for optimisation as show in red areas.

For improving the forecast ability further, the locations of R values in the algorithm can be changed as shown in Table 6.7. The results are listed in Table 6.8.

Table 6. 7 Improvement of R values for deck area of barge model;
Q learning, single objective with two variables;

		A	B	C
		Transverse Bulkhead01	Transverse Bulkhead01	Transverse Bulkhead01
		35	40	45
I	Longitudinal Bulkhead01	20		
	0			
II	Longitudinal Bulkhead01	11		
	5			
III	Longitudinal Bulkhead01			2
	10			

In Table 6.8, the red areas moved to BI and CI and this direction is correct considering the Table 6.3, which indicates that the big deck area should be BI, CI, BII and CII. It is noteworthy that the values in the results are not the prediction of exact deck area but the predictive probability of maximum deck area. In running process, the reward function gives the predictive probability and makes an accumulative calculation. The Q learning is searching the state-action value but not state-state value. This means the optimisation which is searching towards the predictive areas can obtain revenue *maximization*. So if the optimisation is operated according to this direction, the speed of optimisation will be quickened. This demonstrates that the Q learning algorithm (equation (6.12)) can correctly forecast the direction of the maximum deck area.

Table 6. 8 Improvement of the results II for deck area value for barge model ;Q-learning, single objective with two variables;

(a)

		A Transverse Bulkhead01 35	B Transverse Bulkhead01 40	C Transverse Bulkhead01 45
I	Longitudinal Bulkhead01 0		700	762
II	Longitudinal Bulkhead01 5		210	210
III	Longitudinal Bulkhead01 10	375	227	

(b)

		A Transverse Bulkhead01 35	B Transverse Bulkhead01 40	C Transverse Bulkhead01 45
I	Longitudinal Bulkhead01 0		28.18%	30.68%
II	Longitudinal Bulkhead01 5		8.45%	8.45%
III	Longitudinal Bulkhead01 10	15.10%	9.14%	

6.5.3 Case study of box model 2: single objective with four variables

The experiment in this section still focuses on single objective but will extend to a more complex environment. Four variables are employed this time in order to check the performance of Q-learning mechanism (equation (6.12)).

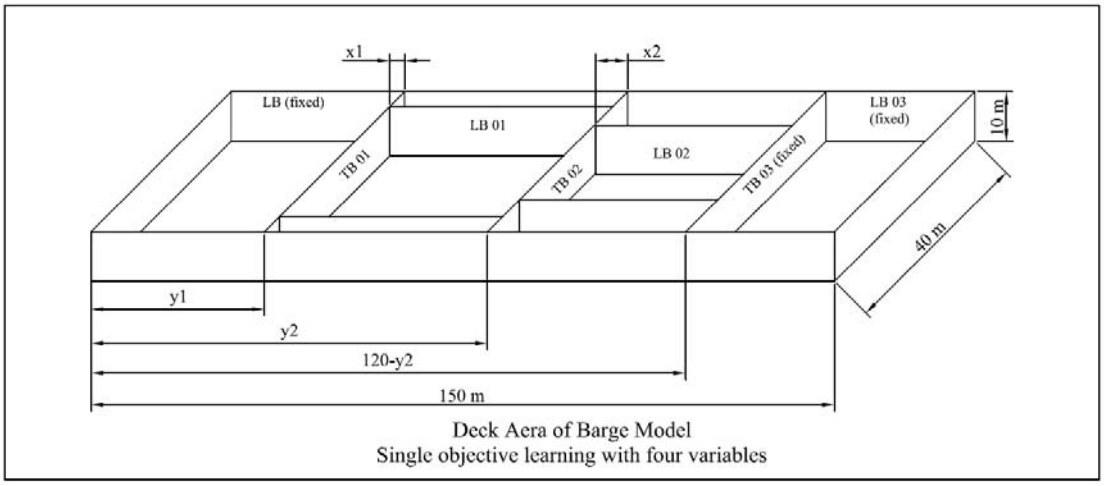


Figure 6. 9 The deck area of barge model, single objective with four variables;

In this experiment, the Transverse Bulkhead TB03 and Longitudinal Bulkheads including LB and LB03 would be fixed. The LB and LB03 are set to zero when TB03 is set to 120m. So the Transverse Bulkheads including TB01 and TB02 with Longitudinal Bulkheads including LB01 and LB02 can be altered. For better understanding, let us set the distance between LB01 and shell as x1 when the distance between LB02 and shell is x2. The distance between the TB01 and ship bow is set as y1 with the distance between the TB02 and ship bow is set as y2. Because location of TB03 is fixed, the distance between the TB03 and ship bow is $120-y2$ as shown in Figure 6.9.

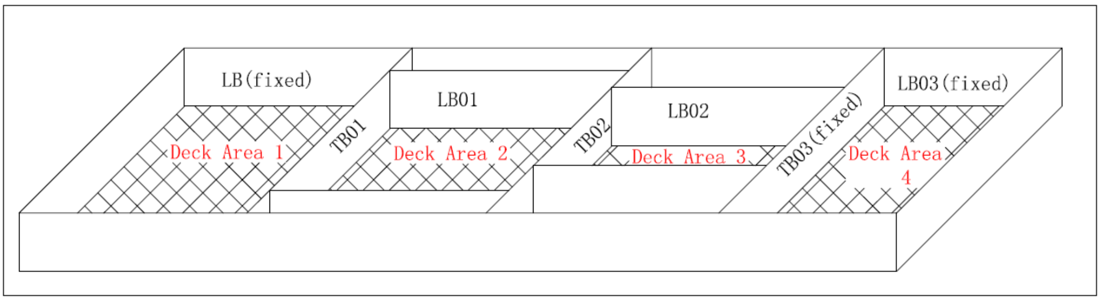


Figure 6. 10 The deck area of barge model

So the deck area of whole barge consists of four parts as shown in Figure 6.10.

Deck area 1 = $y_1 \times 40$;

Deck area 2 = $(y_2 - y_1) \times (40 - 2 \times x_1)$;

Deck area 3 = $(120-y_2)\times(40-2\times x_2)$;

Deck area 4 = $(150-(120-y2)\times 40$;

The problem is

$$\begin{cases} Area_{\max} = y_1\times 40+(y_2-y_1)\times(40-2\times x_1)+(120-y_2)\times(40-2\times x_2) \\ \qquad\qquad +(150-(120-y2)\times 40); \\ x_1, x_2 \in \{0,5,10\} \\ y_1 \in \{35,40,45\} \\ y_2 \in \{75,80,85\} \end{cases}$$

In Table 6.9, there are 81 designs in this case, which are calculated according to real deck areas. The real maximum deck area is represented red as given in Table 6.9. The aim of Q-learning is to find maximum deck areas. In other words, the proposed system should provide the direction towards to red area for optimisation. Correspondingly, 20% random values are given as the initial values. These values are used to calculate R values as shown in Table 6.10. The calculation is processed according to equation (6.12) with $\eta = 0.15$ and $\gamma = 0.95$. The S_{Dstand} selects the average value of all the design value from Table 6.9 and is S_{Dstand}=7000.

Table 6. 9 The real deck area of barge model, Single objective with four variables

No				I	II	III
	x1	x2	y1	y2		
				75	*80*	*85*
1	*0*	*0*	*35*	9000	9200	9400
2	*0*	*0*	*40*	9000	9200	9400
3	*0*	*0*	*45*	9000	9200	9400
4	*0*	*5*	*35*	8550	8800	9050
5	*0*	*5*	*40*	8550	8800	9050
6	*0*	*5*	*45*	8550	8800	9050
7	*0*	*10*	*35*	8100	8400	8700
8	*0*	*10*	*40*	8100	8400	8700
9	*0*	*10*	*45*	8100	8400	8700
10	*5*	*0*	*35*	8600	8750	8900
11	*5*	*0*	*40*	8650	8800	8950
12	*5*	*0*	*45*	8700	8850	9000
13	*5*	*5*	*35*	8150	8350	8550
14	*5*	*5*	*40*	8200	8400	8600
15	*5*	*5*	*45*	8250	8450	8650
16	*5*	*10*	*35*	7700	7950	8200
17	*5*	*10*	*40*	7750	8000	8250
18	*5*	*10*	*45*	7800	8050	8300
19	*10*	*0*	*35*	8200	8300	8400
20	*10*	*0*	*40*	8300	8400	8500
21	*10*	*0*	*45*	8400	8500	8600
22	*10*	*5*	*35*	7750	7900	8050
23	*10*	*5*	*40*	7850	8000	8150
24	*10*	*5*	*45*	7950	8100	8250
25	*10*	*10*	*35*	7300	7500	7700
26	*10*	*10*	*40*	7400	7600	7800
27	*10*	*10*	*45*	7500	7700	7900

Table 6. 10 The R value of real bilge area of barge model with 20% random training sample, Single objective with four variables;

No				I	II	III
	x1	x2	y1	y2		
				75	*80*	*85*
1	*2*	*2*	*35*			
2	*2*	*2*	*40*			34.29
3	*2*	*2*	*45*			
4	*2*	*5*	*35*		25.71	
5	*2*	*5*	*40*	22.14		
6	*2*	*5*	*45*			29.29
7	*2*	*8*	*35*		20.00	
8	*2*	*8*	*40*			
9	*2*	*8*	*45*			
10	*5*	*2*	*35*			
11	*5*	*2*	*40*	23.57		
12	*5*	*2*	*45*			
13	*5*	*5*	*35*		19.29	
14	*5*	*5*	*40*	17.14		
15	*5*	*5*	*45*			23.57
16	*5*	*8*	*35*			
17	*5*	*8*	*40*			
18	*5*	*8*	*45*			18.57
19	*8*	*2*	*35*			
20	*8*	*2*	*40*	18.57		
21	*8*	*2*	*45*			
22	*8*	*5*	*35*		12.86	
23	*8*	*5*	*40*			
24	*8*	*5*	*45*			
25	*8*	*8*	*35*	4.29		10.00
26	*8*	*8*	*40*			
27	*8*	*8*	*45*		10.00	

Table 6. 11 The Results I of Deck Area of Barge Model, Single Objective with four variables

No				I	II	III
	x1	x2	y1	y2		
				75	*80*	*85*
1	*2*	*2*	*35*	64.1	195.1	181.1
2	*2*	*2*	*40*	88.8	169.3	
3	*2*	*2*	*45*	192.5	185	171.9
4	*2*	*5*	*35*	152.4		143.3
5	*2*	*5*	*40*		157	172.8
6	*2*	*5*	*45*	150.7	191.7	
7	*2*	*8*	*35*	139.3		180.4
8	*2*	*8*	*40*	156.2	152.3	122.8
9	*2*	*8*	*45*	61.5	84.3	60.2
10	*5*	*2*	*35*	70.5	31.8	32.5
11	*5*	*2*	*40*		36.6	44.3
12	*5*	*2*	*45*	107.9	76.9	89.4
13	*5*	*5*	*35*	105.7		92
14	*5*	*5*	*40*		111.4	135.9
15	*5*	*5*	*45*	151.9	113.3	
16	*5*	*8*	*35*	57.2	89.6	154
17	*5*	*8*	*40*	58.6	70.8	92.5
18	*5*	*8*	*45*	56.2	132.7	
19	*8*	*2*	*35*	80.6	117.3	117.5
20	*8*	*2*	*40*		79.9	48.4
21	*8*	*2*	*45*	65	69.7	68
22	*8*	*5*	*35*	51		47.6
23	*8*	*5*	*40*	71.8	64.9	61.2
24	*8*	*5*	*45*	22.5	54	43.6
25	*8*	*8*	*35*		46.9	
26	*8*	*8*	*40*	47.1	28.2	30
27	*8*	*8*	*45*	44.5		44.6

As shown in Table 6.11, the proposed method calculate the predictive value for optimisation and for better understanding, the Table 6.11 will be transferred to Table 6.12 with percentage format.

Table 6. 12 The Results II of Deck Area of Barge Model, Single Objective with four variables

No	x1	x2	y1	I	II	III
					y2	
				75	*80*	*85*
1	*2*	*2*	*35*	1.00%	3.05%	2.83%
2	*2*	*2*	*40*	1.39%	2.65%	
3	*2*	*2*	*45*	3.01%	2.90%	2.69%
4	*2*	*5*	*35*	2.39%		2.24%
5	*2*	*5*	*40*		2.46%	2.70%
6	*2*	*5*	*45*	2.36%	3.00%	
7	*2*	*8*	*35*	2.18%		2.82%
8	*2*	*8*	*40*	2.44%	2.38%	1.92%
9	*2*	*8*	*45*	0.96%	1.32%	0.94%
10	*5*	*2*	*35*	1.10%	0.50%	0.51%
11	*5*	*2*	*40*		0.57%	0.69%
12	*5*	*2*	*45*	1.69%	1.20%	1.40%
13	*5*	*5*	*35*	1.65%		1.44%
14	*5*	*5*	*40*		1.74%	2.13%
15	*5*	*5*	*45*	2.38%	1.77%	
16	*5*	*8*	*35*	0.90%	1.40%	2.41%
17	*5*	*8*	*40*	0.92%	1.11%	1.45%
18	*5*	*8*	*45*	0.88%	2.08%	
19	*8*	*2*	*35*	1.26%	1.84%	1.84%
20	*8*	*2*	*40*		1.25%	0.76%
21	*8*	*2*	*45*	1.02%	1.09%	1.06%
22	*8*	*5*	*35*	0.80%		0.75%
23	*8*	*5*	*40*	1.12%	1.02%	0.96%
24	*8*	*5*	*45*	0.35%	0.85%	0.68%
25	*8*	*8*	*35*		0.73%	
26	*8*	*8*	*40*	0.737%	0.441%	0.470%
27	*8*	*8*	*45*	0.697%		0.698%

In Table 6.12, the red area provides the direction of optimisation, which means these red areas give the quickest way to arrive the maximum area. Comparing with Table 6.9, the proposed approach can correctly provide the direction of optimization. So the optimisation should go to this direction for fast convergence.

In this study the GA developed in JAVA language is deployed to deal with the same problem as the one given in section 6.5.3; '*case study of box model 2: single objective with four variables*'. Two algorithms were in the same computer environment with the same parameter settings. Then every algorithm was run 10 times. The program searches the solutions until it finds the eight solutions as given in Table 6.9 (70%). The average time is calculated to compare the speed of algorithm. The GA without Q-learning takes 211 seconds and GA with Q-learning takes 156s which means that the GA with Q-learning improve the computation time by 26%.

6.6 Discussion

In this chapter, the theory and application of ship real-time learning in optimisation process were explained. As a case study, the application of Q learning is demonstrated using the barge model. The results indicate that the Q learning can improve optimisation and reduce running time. The proposed system is tested and validated successfully. The learning model is created from the brain science and combined with the ship design practice. Q-learning, as a detailed approach of sensory memory part, provides the advantages for realizing the learning function.

The proposed algorithm is structured via a multi-agent system and every agent worked remarkably well. It can be concluded that the proposed system has shown great potential and can be applied to similar and even more complex optimisation problems in ship design as well as to related areas within the maritime industry.

Chapter 7

Learning Based Decision Making and Decision Support System in Ship Design

7.1 Introduction

The learning based decision making is one of the key parts of this ship design decision system proposed in this thesis. After the multi-objective optimisation is completed and the Pareto feasible solutions are given, it is critically important to know how to identify the most suitable solutions from the optimal solutions. The numerical calculations of optimisation provide a series of solutions to the designers as a solid support but the support system needs the further analysis to make an accurate and scientific decision as the final solution. At this stage of the design, the opinions including the linguistic attribute of different specialist will be comprehensively considered. So a powerful and intelligent method is required to help the designers to make a good decision.

A fuzzy multiple attribute decision-making (FMADM) method, which can solve both linguistic and numerical attribute, is introduced here to solve the decision making problem. The linguistic attribute is one of the most difficult aspects in decision making period. In the proposed method, the specialist committee decide the quality of design, which totally depends on individual's knowledge and experience level. Then the technology manager, who allocates the weighting, further improves the interaction with human. This study proposes a new learning based virtual specialists committee which can use prior experience to evaluate the solutions. It also creates relevant virtual technology manager, who will allocate the weighting to every member of the committee

For better application in ship decision support system, an agent based framework is utilised to realize this method in computer environment. The method is rebuilt according to module based design principle, which makes the system satisfy the change of designer's requirement and the expansion of the data. A case study on subdivision is used to evaluate the method.

7.2 Problem Definition

In decision based ship design, decision making is a very important problem for whole ship design. Ölçer et al. (Ölçer, Tuzcu et al. 2006) successfully employed a fuzzy multiple attribute decision-making (FMADM) method to solve the subdivision design of a passenger ship. But this FMADM method has two problems in application. The first one is that the realization of this approach in computer environment depends on manual controlling and Excel forms, in other words, the approach is semi- automatic. This makes the approach unsuitable for complex decision making environment under certain conditions. The second one is that the specialists and technology manager play the key role in this approach, which makes the decision process greatly dependent on the level of human taking part in the evaluation process. This approach cannot store the experience of event and relies on the personal ability of specialists who should be involved in every run. This reduces the robustness of the system, which means that slight difference in weighting will probably cause significantly different final decision.

The first problem creates complex operations for designers. The designers have to make frequent switch between Excel forms and manual calculation. This is difficult for the designers without decision making knowledge. At the same time, the ranking approach is also difficult to understand for new learners. The second problem is the time it takes to reach a decision. If there are not enough specialists, the decision making can not be processed. Even if the designers bring enough number of specialists together, the evaluation still needs a long time.

This chapter will focus on solving these two problems via machine learning approaches. The new multi-agent frame is employed to realize the FMADM approach, which makes this approach more convenient for embedded system. What is more, the multi-agent system makes the approach more modular and user can freely replace any parts of the approach, for example, original approach use TOPSIS approach for ranking, and user can choose the other more efficient new ranking approaches. This can be realized by just replacing the ranking module without changing the other modules.

For the second problem, the virtual specialist committee and technology manager are created to avoid the absence of human being. The original method needs the specialist committee to give an evaluation. But in normal situation, this requirement is difficult to implement in a short time and also the evaluation would take a long time. What is more important is the specialist committee does not have inheritance, which means if there are not enough people to make the specialist committee, the method cannot be processed. Due to the lack of prior information to share for similar past problems, the prior experience can not be used for next time. If the prior experience can be utilized, the system can give a predictable expert simulation. Therefore building a machine learning based virtual specialist committee is a good improvement for the stability and robustness of this FMADM method.

The virtual specialists committee can give an evaluation based on prior examples to replace the human specialists. In the beginning, the human specialists are convened to evaluate the samples. The system saves these evaluations and makes an analysis via Support Vector Machine (SVM) learning approach. After finding the relationships between the data and evaluations, the system can automatically judge the situation with the new cases and give the correct evaluation. So, the decision making method does not need human specialists during the run and can give evaluation once the nearest experience is found in the software.

In summary, there are two contributions in this chapter: one is the new Multi-agent based FMADM (MFMADM) method for decision making, which combines the multi-agent theory and FMADM. The second is the new Learning based MFMADM (LMFMADM). LMFMADM add the learning function to MFMADM.

For rebuilding of multi-agent framework, the most important thing is to make clear the functions and context of every agent. This requires transferring the original algorithm to independent agent, which can deal with the sub-jobs successfully and also can collaborate together to solve the whole problem. The other important thing is how to define the communication and conflict resolution in the application.

For learning based virtual specialists and technology manager, the research focuses on the selection of an appropriate method to give precise forecast result. Because the behaviours of specialists and technology manager greatly affect the final decision, the learning method in this part should be more accurate.

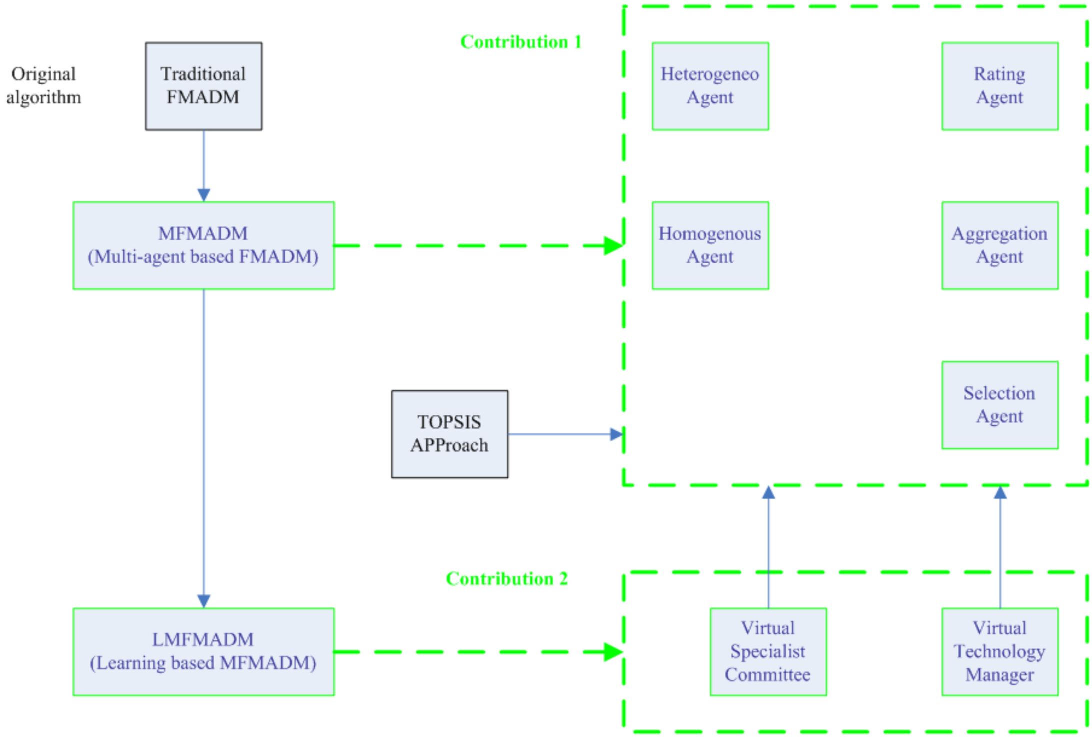


Figure 7. 1 The structure of chapter 7

This chapter is organized according to Figure 7.1. The traditional FMADM is the foundation of new developed multi-agent based FMADM, which contains two parts. The first one is multi-agent based FMADM. This new developed FMADM makes two main contributions. One is that the new method transfers operation from traditional semi-automatic to new automatic programme. One of the advantages of this modification is that the automatic programme provides great convenience for integrated decision support system. Another contribution is that the modularisation will bring fast updating of technology.

The second innovation part is the virtual committee, which again contains two sub-section. One is specialist committee, which will evaluate the results from optimisation part. Another is the technology manager, which is responsible for weighting distribution. In this part, a new learning approach — SVM will be introduced and used to build this committee.

The objectives of rebuilding of multi-agent framework are as follows:

1. Redefine the agent based workflow;

2. Define the agents and build the multi-agent system;
3. Introduce the Support Vector Machine (SVM) in this system;
4. Build the virtual specialist and technology manager;
5. The training run of virtual specialist and technology manager;

7.3 Learning and Multi-agent based FMADM (LMFMADM)

The fuzzy multiple attribute decision-making (FMADM) method is proposed by Olcer and Odabasi (Ölçer 2001). It is suitable for multiple attributive group decision making (GDM) problems in fuzzy environment, and has been employed to deal with some ship design decision problems (Ölçer, Tuzcu et al. 2005), (Ölçer 2008). In this section, an agent based frame in computer environment is taken into account, which can greatly improve the calculation efficiency compared to the original Excel based partial auto-calculation.

The contributions of new learning based FMADM concludes two parts. The first one is the revision of FMADM from Excel based partial auto-calculation to multi-agent based auto-calculation. This contribution improves the feasibility of the algorithm. The second one is to add the learning ability to FMADM. The learning ability solves the problem of lacking of experts and improves robustness of the algorithm.

7.3.1 Basic FMADM method

Olcer and Odabasi (Ölçer 2001) reviewed and analyzed the most of the known FMADM methods according to their group decision-making ability. Based on this research, they provide a new FMADM approach that can be utilised in ship design, for example, propulsion/manoeuvring system selection or subdivision optimisation.

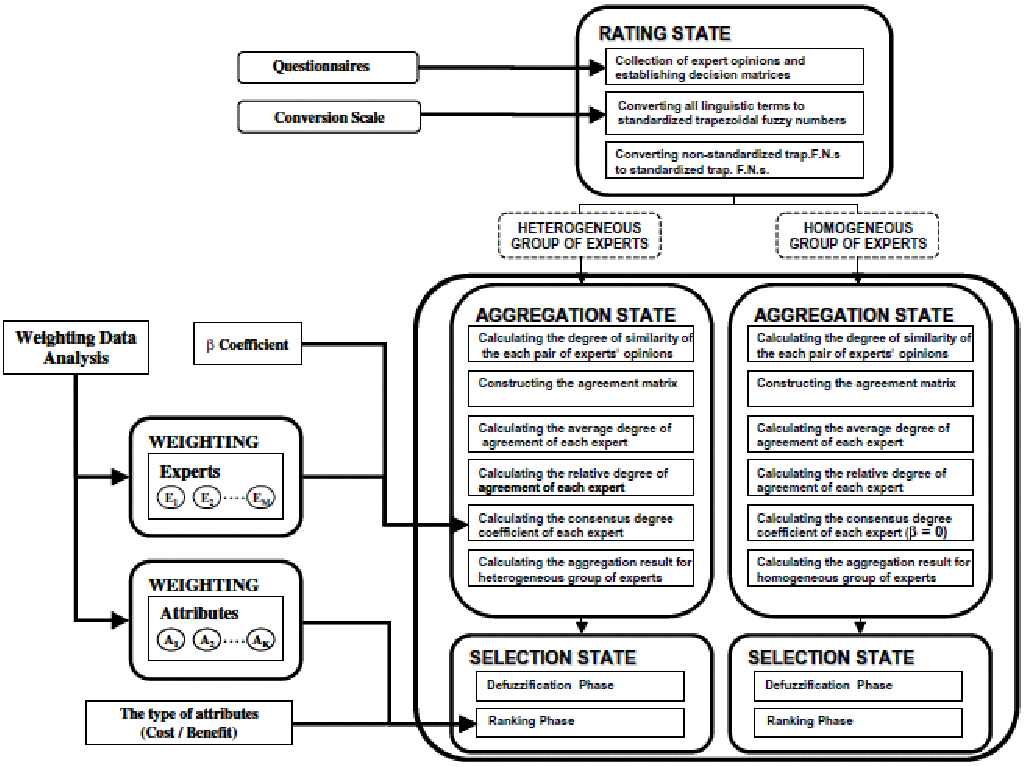


Figure 7. 2 a the work flow of FMADM used in this study

(This graph is taken from Olcer etc. 2001)

The authors have given the work flow as shown in Figure 7.2 a. In order to develop learning based FMADM, multi-agent system has to be modified as the basic step. A multi-agent based work flow is developed to make a clear understanding of whole rebuilding process of this approach as shown in Figure 7.2 b.

7.3.2 New LMFMADM method

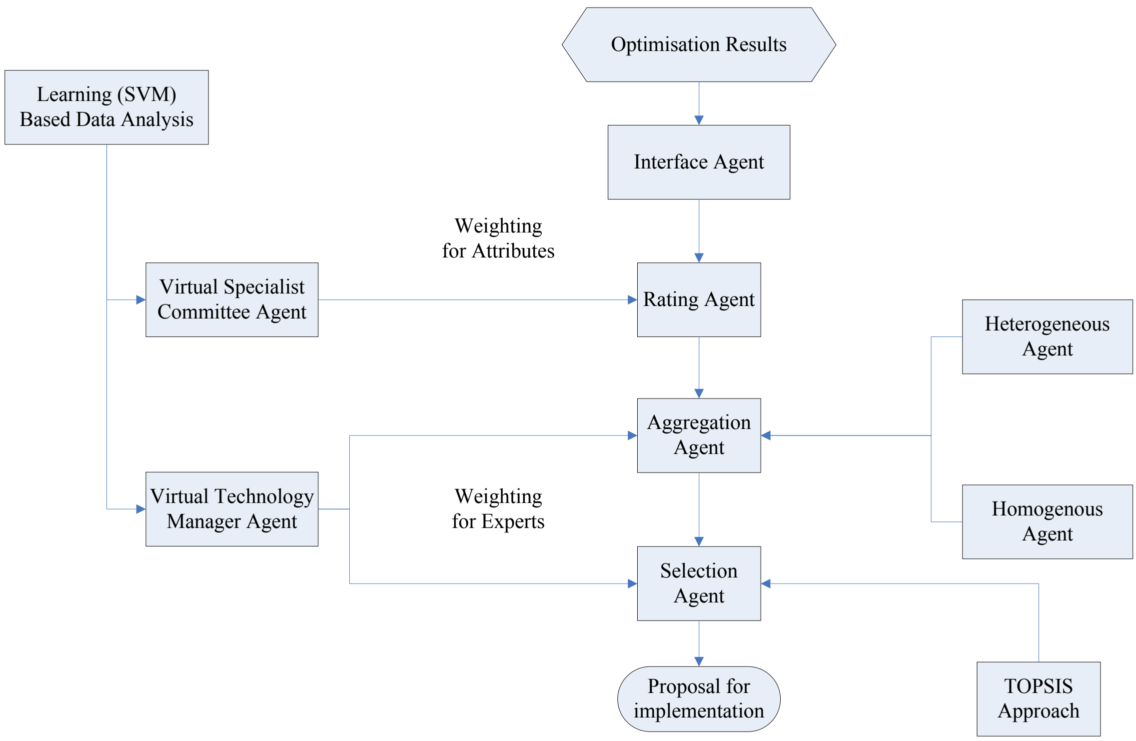


Figure 7.2 b Work flow of new MFMADM

In Figure 7.2 b, there are seven software agents and two human agents. A standard agent architecture, as shown in Figure 7.3, which was employed in previous work (Turkmen 2005), is also utilised here to construct the agent.

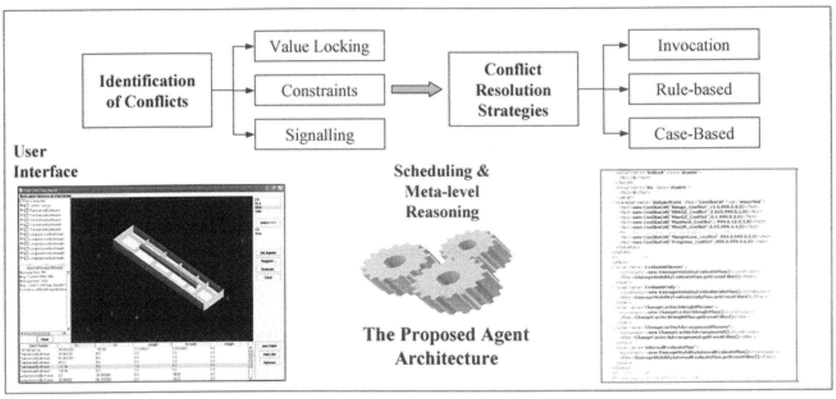


Figure 7. 3 the proposed intelligent agent architecture and conflict resolution (taken from the thesis of Turkmen 2005)

In the intelligent agent architecture of Figure 7.3, the agent is divided into different layers and for some layers, there are sub-layers. This study follows the organization of Figure 7.3 and improves the sub-layer structure. The new agents are also created according to proposed decision making method. The detailed functions and operations of every agent are developed and realized in the system.

In following, every agent and its layers are explained.

Interface Agent: The function of this agent is to transfer the Pareto-optimal designs, obtained from the optimisation part, to an available format of this decision making process. This agent is the simplest agent in this approach and the aim is to read data, classify and calculate the number of attributes.

Communication Layer: This is the main part of this agent. The agent will receive a signal from the optimisation module and open the 'door' for the data transfer.

Negotiation/Collaboration Layer: Because this is a simple agent, there is only one module, acquaintance module, being active. This acquaintance module lists all of the agents and stores the data for different agents.

Task Layer: The task layer of this agent has two functions. One is the input function and other is the counter function. The input function reads the data and corrects the format. Then it will store the data according to the usage of every agent like a classification system with database. The counter function is responsible for calculating the attributes to construct the matrix of following agents.

Rating Agent: The aim of this agent is to integrate fuzzy data into standardised positive trapezoidal fuzzy numbers and establish the decision matrix.

Communication Layer: The communication layer is to connect one software agent with one human agent. The software agent is interface agent, which gives the initialized data including optimisation solutions to this agent. The human agent is specialist committee agent, which will evaluate of optimisation solutions.

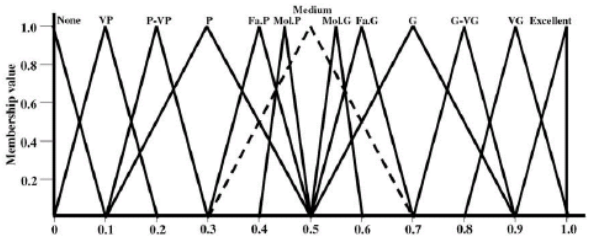
Negotiation/Collaboration Layer: This layer contains several sub-layers. The first sub-layer is *acquaintance module* sub-layer. There are two parts in this sub-layer. The first one is the list of agents, which is necessary for controlling the actions of every agent in the multi-agent system. The other one is the agent services, which also includes two parts: the agents which act on this agent and the agents which this agent will act on. The second sub-layer is *conflict resolution module* sub-layer. This sub-layer uses the rules to check the validity of the solutions coming from the task layer. The last sub-layer is optimisation module and learning module. In original algorithm, there is no application of these two modules. But following the rebuilding with learning ability, they will be used. So here the structure is kept some as before but not the context.

Task Layer: The task layer of this part will not employ third party software but to make its own coding via Java language. One of the difficult points is the conversation of fuzzy data. Normally, the decision matrix will contain large amount of fuzzy data because that the linguistic terms are more easy to express the opinions of specialists. For example poor and good make sense, but {0 1}, {4 5} give no idea both to the specialist and to the system. The similar Linguistic terms and their corresponding fuzzy numbers and membership functions are used as in Olcer et al. (Olcer et al. 2005) as shown in Figure 7.4. All the linguistic terms will be transferred to numbers. The output of this agent is a numerical matrix.

Linguistic terms and their corresponding fuzzy numbers and membership functions used in the proposed approach

	Linguistic terms	Abbreviation	Corresponding Fuzzy Numbers
1	None	None	Trap (0, 0, 0, 0.1)
2	Very Poor	VP	Trap (0, 0.1, 0.1, 0.2)
3	Poor – Very Poor	P-VP	Trap (0.1, 0.2, 0.2, 0.3)
4	Poor	P	Trap (0.1, 0.3, 0.3, 0.5)
5	Fairly Poor	Fa.P	Trap (0.3, 0.4, 0.4, 0.5)
6	More or Less Poor	Mol.P	Trap (0.4, 0.45, 0.45, 0.5)
7	Medium (also Fair)	Medium	Trap (0.3, 0.5, 0.5, 0.7)
8	More or Less Good	Mol.G	Trap (0.5, 0.55, 0.55, 0.6)
9	Fairly Good	Fa.G	Trap (0.5, 0.6, 0.6, 0.7)
10	Good	G	Trap (0.5, 0.7, 0.7, 0.9)
11	Good – Very Good	G-VG	Trap (0.7, 0.8, 0.8, 0.9)
12	Very Good	VG	Trap (0.8, 0.9, 0.9, 1)
13	Excellent	Excellent	Trap (0.9, 1, 1, 1)

Membership functions of fuzzy numbers used in the proposed approach



Where Trap indicates a standardised trapezoidal fuzzy number

Figure 7. 4 Linguistic terms and their corresponding fuzzy numbers and membership functions (This graph is taken from Olcer etc. 2005)

Aggregation Agent: The aim of this agent is to combine the opinion of single or multidiscipline specialist to form a group consensus opinion. There is close linkage between this agent and technology manager.

Communication Layer: This communication layer is also used to connect one agent and one human interface. The agent is the rating agent and the human interface is technology manager, who allocates the weightings of specialists.

Negotiation/Collaboration Layer: This layer contains two sub-layers. Firstly, the acquaintance module sub-layer contains the list of other agents and information on the needs of these agents. The second one is the conflict resolution module to check the validity of task layer.

Task Layer: The task layer of this agent will employ other two agents, the homogeneous agent and the heterogeneous agent. So it just contains the code to call other agents and there is no calculation part.

***Heterogeneous/Homogeneous Agent*:** the aim of this agent is to give the result of the fuzzy opinions. It needs to measure the degree of similarity between trapezoidal fuzzy numbers.

Communication Layer: This communication layer connects only to: Aggregation Agent. It will read data from Aggregation Agent and the results will be given to the Aggregation Agent. So this agent just talks with Aggregation Agent.

Negotiation/Collaboration Layer: This layer is similar with the Aggregation Agent.

Task Layer: The task layer of this agent uses JAVA language for coding. It will not employ other third party software.

Selection Agent: the aim of this agent is to select the best solutions according to the suggestions of specialists. It will employ a ranking approach to give the rank of suggestions.

Communication Layer: The communication layer connects to three agents, which are Aggregation Agent, TOPSIS approach agent and technology manager. This agent reads the matrix from the aggregation agent then will input the weightings from the technology manager while employing the TOPSIS approach agent to rank.

Task Layer: The task layer of this agent selects the final solution according to the specialist's option. So it needs to analyses the suggestions from specialists and the ranking results from TOPSIS agent.

TOPSIS Agent: This agent is a special approach agent. Here it does not employ the normal structure but uses the work pattern. This means this agent does not employ other agents and can not be interfered by other agents. It is jut a pure work agent.

Technology manager and ***Specialist Committee*** are the external human interfaces. They are responsible for evaluating the optimisation results.

It can be seen that in this approach, the technology managers and specialists are very important to engineering application. Every time, designers have to organize a committee and manager to discuss the optimisation solutions. But organizing a high level committee is not easy. The final decision is decided by the knowledge level of committee and manager. In order to improve the robustness of decision, a viral experience based committee and manager are proposed and SMV method is employed in order to realize this idea.

There are three main advantages of the rebuilding. The first point is the rebuilding change from manual/semi-automatic to automatic. The automatic improves the efficiency of this method and also transfers it to a good decision making tool for practical engineering application. The designers can directly use this method without the special knowledge and the method would give the selections and reasons.

The second advantage is that module based rebuilding makes the update easier. For example, the original method introduced one ranking approach---TOPSIS. With the development of ranking theory and application, the new and more powerful ranking approaches continue to come out. If the designers want to update a new ranking approach, they just need to replace the old ranking agent. It is easy to realize without changing other agents.

The last advantage is that the rebuilding adds the convenience for improving original method. In the following part, an experience based learning approach will be integrated in FMADM to improve the ability of absence specialists. In original method, the system has to make a linking to every part. After rebuilding, the system just needs to add a new agent with standard input and output.

7.4 The machine learning method used in this chapter

There are several machine learning approaches proposed in previous chapters but a new one will be introduced here to assist the system to realize the learning function specific for the problems addressed in this chapter. This chapter requires the learning method which can make an accurate prediction according to small sample. At the same time, the learning method should have the excellent ability to control the complex learning environment while providing a clear explanation to the users. The Support Vector Machine (SVM) as a powerful machine learning approach is selected here to deal with above learning problem during decision making stage.

"The Support Vector Machine (SVM) is a supervised learning method that generates input-output mapping functions from a set of labelled training data. The mapping function can be either a classification function, i.e., the category of the input data, or a regression function", (Wang 2005). Generally, the 'machine' in SVM is not a real machine. In machine learning, an algorithm is always called machine, so SVM continues using this custom and 'machine' here means algorithm. The word 'support vector' comes from the training samples in SVM which are expressed via vectors and SVM strongly focuses on the vectors at the edges which support the seeking of hyper plane. Usually, a support vector machine is constructed by a hyper-plane or set of hyper-planes in a high or infinite dimensional space. The hyper-plane, which has the largest distance to the nearest training data points of any class, can make a good separation to data. In this section, firstly, the basic concept of SVM including the extension statistical learning definition will be introduced. The extension statistical learning definition consists Empirical Risk Minimization principle (ERM), Structure Risk Minimisation (SRM). Then, on above foundation, a general application for building virtual committee is introduced. Finally, combined with practical applications, the virtual committee including specialists and technology manager is built.

7.4. 1 The theory of Support Vector Machine (SVM)

The learning function at this stage has several critical properties which can greatly affect the selection of the learning method. Firstly, the decision making process must be controlled strictly by decision maker. This means that the supervised learning is more appropriate and a friendly human-computer interaction system is needed.

Secondly, the training set is very small and the continuing development of training set is also limited. At this stage, the specialists in real world give the professional evaluation to every design. The numbers of specialists are limited and the cases, which are evaluated, are also limited and have recalling the specialists is very difficult. So the learning method should have good ability to operate the small training set at this stage. The third one is the precision. This part needs the learning method to give the accurate prediction. The precision requirement of this part is significantly higher than the methods presented in previous chapters because the slight variation will cause great difference in the final results. The last one is the explicitness. This is also a basic principle for the whole support system. For this point, especially for small samples, in order to insure the precision of the results, a learning method with solid mathematical foundation is needed.

The support vector machine is a kind of novel machine learning method to solve the nonlinear and multi dimensional problem with small sample sets. The main ideas are developed in 1990s, and it has the firm foundation of mathematical theory.

First of all, a brief explanation of the task of machine learning will be given here to clear the aim of employing the support vector machine. Without the loss of generality, let x be an input of one system and y be the output of this system. Normally, it can be considered that there is a certain function which can express the relationship between x and y. So the task of machine learning is to seek an optimum function to represent this relationship as far as possible according to training sets $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, which means that the estimation error of this function is minimum. In other words, ultimately the expected risk function (Equation (7.1)) should be minimized. (Vapnik 1995)

$$R[f] = \int_{X \times Y} L(y, f(x)) P(x, y) dxdy \tag{7.1}$$

In equation (7.1), $P(x, y)$ is the joint probability distribution function which represents the relationship between x and y. $L(y, f(x))$ is the loss or discrepancy between the response y of the supervisor to a given input x and the response $f(x, \alpha)$.

But the expected risk function cannot be calculated by depending only on the training set, which is usually small sample data. So in statistical learning theory, the ***Empirical Risk Minimization principle (ERM)*** is accepted.

$$R_{emp}[f] = \frac{1}{l}\sum_{i=1}^{l} L(y^i, f(x^i)) \qquad (7.2)$$

The task of machine learning is transferred to design an algorithm to minimize Equation (7.2). It is noteworthy that the ERM is not strictly proven in mathematics, but it is intuitive and has a leading position in machine learning for a long time. Obviously, this method has theoretical defect and a classic example is ANN (Artificial Neural Network). For ANN, the training error can be very small in training sets but the results may be bad when this ANN is applied to new cases, which is called over learning in the machine learning area. The reason of this situation is that the ANN can remember every training set but can not improve to give a good prediction for new inputs, in other words, the system tries to find a complex model via limited samples and the model, which was developed by the system, has the excellent adaptability for training sets but poor generalization for new cases. So a theory which can make learning from small samples is fully necessary.

Before discussing the new algorithms, some definitions should be explained for a better understanding. The first concept is ***VC (Vapnik–Chervonenkis) dimension***. The VC dimension is a scalar value that measures the capacity of a set of functions. The VC dimension of a set of functions is p, if and only if there exists a set of points $\{x_i\}_p$ such that these points can be separated in all 2p possible configurations, and that no set $\{x_i\}_q$ exists where q > p satisfying this property (Steve R. Gunn). The VC dimension reflects the capacity of learning ability, and larger VC dimension means the learning is more complex.

The second concept is "***Bounds on the Generalization Ability of Learning Machine***". This concept is complex and here simply the result of the concept is introduced:

The following bound in Equation (7.3) holds with probability $1-\eta$,

$$R[f] \le \mathrm{R}_{emp}[f] + \sqrt{\frac{h\ln(\frac{2n}{h}+1)-\ln(\frac{\eta}{4})}{n}} \tag{7.3}$$

$R\left[f\right]$ is the actual risk, n is the number of training sets and h is VC dimension. The actual risk which is related to VC dimension and the number of training sets, is divided into two parts: the empirical risk and confidence interval. So the machine learning should make minimization both on the empirical risk and confidence interval, which means the VC dimension, should be as small as possible to minimize the actual risk in order to obtain good generalization ability in the future.

The third concept is ***Structural Risk Minimisation (SRM)***, which aims to minimize both the empirical risk and the confidence interval.

$$\min \mathrm{R}_{emp}[f] + \sqrt{\frac{h\ln(\frac{2n}{h}+1)-\ln(\frac{\eta}{4})}{n}} \tag{7.4}$$

Here, the VC dimension is made as a controlling variable. For better understanding, the set S (of function $Q(z,\alpha), \alpha \in \Lambda$), which is structured via nested subsets of functions, is given. $S_k = \{Q(z,\alpha), \alpha \in \Lambda_k\}$:

$S_1 \subset S_2 ... \subset S_n ...,$

where the elements of the structure satisfy the following two properties:

(i) The VC dimension h_k of each set S_k of the function is finite

(ii) Any element S_k of the structure contains either

A set of totally bounded functions

$0 \leq Q(z,\alpha) \leq B_k, \alpha \in \Lambda,$

or a set of functions satisfying the inequality

$$\sup_{\alpha \in \Lambda_k} \frac{(\int Qp(z,\alpha)dF(z))^{\frac{1}{p}}}{\int Qp(z,\alpha)dF(z)} \leq \tau_k, p > 2, \qquad (7.5)$$

For a pair (p, τ_k), p has defined in VC dimension concept and τ_k is a real number. For a given set of observation $z_1, ..., z_l$, the SRM principle chooses the function $Q(z, \alpha_l^k)$ minimizing the empirical risk in the subset S_k for which the guaranteed risk, which is determined by the empirical risk and the confidence interval, is minimal.

Then, come to the support vector machine. The support vector machine designs the set S to make every subset obtain the minimal empirical risk and selects the appropriate subset whose confidence interval is minimal.

In the following parts of this section, the support vector machine is introduced briefly. The detailed theory and mathematical proof are given by Vapnik (Vapnik 1998).

Without loss of generality, consider the situation that there are two types of samples which need to be classified as shown in Figure 7.5.

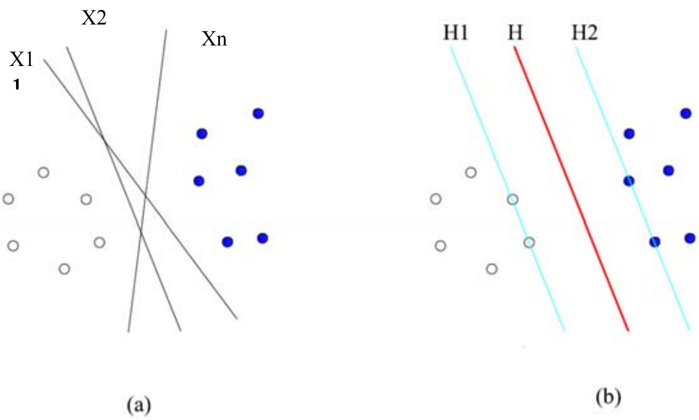


Figure 7. 5 Two sample Sets need to be classified

There are many possible linear planes (X1, X2,…,Xn) that can separate the data as shown in Figure 7.5 (a). In order to find the hyper plane, the first step is to define the distance between two samples and the nearest data point of other class sets as margin, for example, the distance between H1 and H2. Then the optimal separating hyper plane (H) is defined as the linear classifier which maximizes the margin.

Consider the problem of separating the set of training vectors belonging to two separate classes as Equation (7.6):

$$D = \{(x^1, y^1), \ldots, (x^l, y^l)\}; \quad x \in R^n, y \in \{-1, 1\} \tag{7.6}$$

In Equation (7.6), D is the set which represents the whole space. The x is the location of point and y represent the class of the point.

A separating hyper plane in canonical form must satisfy Equation (7.7).

$$y^i [\langle w, x^i \rangle + b] \geq 1, i = 1, \ldots, l. \tag{7.7}$$

This constraint is that the norm of the weight vector should be equal to the inverse of the distance of the nearest point in the data set to the hyper plane.

It can be proven that the hyper plane that optimally separates the data is the one that minimizes the distance between the edges of different samples. (as shown in Equation (7.8)):

$$\Phi(w) = \frac{1}{2}\|w\|^2 = \frac{1}{2}\langle w, w \rangle. \tag{7.8}$$

So the classifier, which minimizes Equation (7.8), is the hyper plane. In Figure 7.5, this classifier is H. The sample sets on H1 and H2 are called support vector, which is the reason of support vector machine (The 'machine' in machine learning sometimes means the algorithm).

Through Lagrange method, Equation (7.8) can be transferred to

$$Q(\alpha) = \sum_{i=1}^{n} \alpha - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle, \tag{7.9}$$

Equation (7.10) is obtained by solving Equation (7.9),

$$f(x) = \mathrm{sgn}(\langle w^*, x \rangle + b); \tag{7.10}$$

$$w^* = \sum_{i=1}^{l} \alpha_i y_i x_i; \quad b^* = -\frac{1}{2}(w^*, x_r + x_s);$$

Form Equation (7.8), (7.9) and (7.10), it can be seen that SVM is concerned with inner product. An inner product in feature space has an equivalent kernel (Equation (7.11)) in input space,

$$K(x, x') = \langle \phi(x), \phi(x') \rangle \tag{7.11}$$

Therefore, if a suitable kernel function, which is corresponding to an inner product in a transformed space can be utilised, then the SVM can be transferred to the function based on this kernel function. In SVM, using kernel function can realize the mapping from low dimension space to high dimension space without adding calculation complexity. In other word, the kernel function can help SVM to realize to transfer nonlinear classification to linear classification without adding calculation complexity.

There are several types of kernel functions and three of them are studied broadly. A polynomial mapping is a popular method for non-linear modelling as Equation (7.12),

$$K(x,x') = (\langle x,x' \rangle + 1)^d . \tag{7.12}$$

Gaussian Radial Basis Function is Equation (7.13). Radial basis functions have received significant attention, most commonly in a Gaussian form.

$$K(x,x') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right) \tag{7.13}$$

When the SVM is applied in practice, the user can select the appreciate kernel function for the problem. In this system, these three methods namely Lagrange method (Equations (7.10)), Polynomial mapping method (Equations (7.12)) and Gaussian radial basis function (Equations (7.13)) are employed for comparing the utility of different kernel functions.

In this section, the concepts of SVM and static learning including VC dimension, ERM, SRM are introduced together with the theory of SVM. This section also presents the kernel function and the reason to employ in SVM.

7.4.2 The general method of SVM application in FMADM

In section 7.4.1, the concepts and theory of SVM has been explained. In this section, the detailed application procedures will be presented to solve the practical problem. Taking into account the complexity of practical problem in ship design, the procedures are introduced from linear to nonlinear step by step.

7.4.2.1 The linear method in SVM for application

The first approach is the linear method. Although the nonlinear classification has become popular research field, the linear programming still have great advantage due to the simplicity of algorithm and easy parameter setting. When a new problem is coming, the linear classification should be preferred as the first choice.

The linear method defined here means that if a linear function can correctly classify the sample space, this sample space is named as linearly separable and the method to find the linear function is named as linear method. If not, the sample space is nonlinearly separable and the method is nonlinear method.

Assume the linear function has the form as shown in Equation (7.14):

$$g(x) = (w \cdot x) + b; \qquad (7.14)$$

If Equation (7.14) is used as a decision function, a threshold should be defined to distinguish different categories. For example, if the threshold is defined as 0, the samples, which are less than 0, belong to category 1 and others belong to category 2. Then if g in Equation (7.15) is less than 0, the sample is classified as category 1 and if not, the sample is classified as category 2. In mathematics the above process equals to appointing a sign function to Equation (7.15).

So the decision function is:

$$y = \mathrm{sgn}(g) = \mathrm{sgn}((w \cdot x) + b) \qquad (7.15)$$

w and b are vectors.

It is noteworthy to highlight if Equation (7.15) does not have to be limited to two dimensions. For n dimension problem, the w and b are n dimension vectors. So in linear method, Equation (7.15) is the hyper plane to classify the data. The aim of linear method is to find solutions of w and b. The whole process can be described as stylization procedure (a) (Deng and Tian 2004).

<u>Stylization procedure (a):</u>

Step 1 Assuming that the sample is

$T=\{(x_1,y_1),...,(x_l,y_l)\}\in(X\times Y)^l$, $x_i\in X=\mathbb{R}^n, y_i\in Y=\{1,-1\}, i=1,...,l;$

Step 2 Construct and solve (7.16)

$$\begin{aligned} &\min \quad \Phi(w)=\frac{1}{2}\|w\|^2, && \text{(a)} \\ &s.t. \quad y_i((w\cdot x_i)+b)\geq 1, \quad i=1,...,l, && \text{(b)} \end{aligned} \qquad (7.16)$$

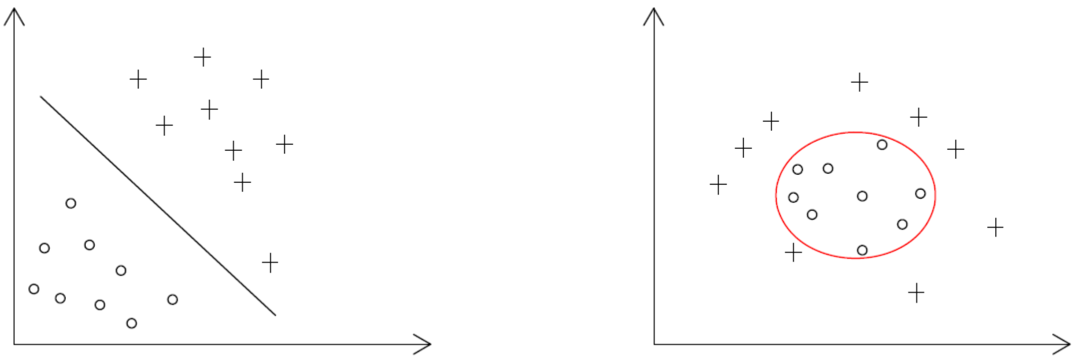
and obtain the solution w^* and b^*.

Step 3 Construct hyper plane $(w^*\cdot x)+b^*=0;$, and obtain decision function: $f(x)=\text{sgn}((w^*\cdot x)+b^*)$

7.4.2.2 The nonlinear method in SVM for application

In above procedure, the samples must be linearly separable, but in most of the time, the samples may be only nonlinearly separable. So this procedure should be extended to nonlinear cases. At the same time, above classification problem has only two categories. In the practical ship design problem, the classification problem usually is multi-category, which means the categories are often more than two, for example, in Appendix A, the cargo capacity has four categories including normal, good, very good and excellent. So this procedure also should be extended to more general situation based on the number of categories.

Firstly, the procedure will be extended to nonlinear situation and then in order to have better understanding of what nonlinearity is, a two dimensional example is given.



(a) (b)

Figure 7. 6 Two dimensions example of nonlinear classification

In Figure 7.6 (a), two kinds of data can be classified by a line, but in the situation of Figure 7.6 (b), the data can not be classified by a line. The sample space in Figure 7.6 (a) is linearly separable space while the sample space in Figure 7.6 (b) is nonlinearly separable space.

For nonlinearly separable space, an important method to solve classification problem is to find an effective transformation from nonlinear to linear. In Figure 7.6 (b), an ellipse can be used to separate the data. So if the line can be transferred to ellipse or other forms, the nonlinear problem can be solved.

In order to take into account the procedure of above linear example which is spontaneously formed, a stylization procedure is necessary for better understanding the difference between the linear and nonlinear. The above linear classification of SVM can be seen as a minimum solution of Equation (7.16):

Actually, Equation (7.16) consists of Equations (7.7) and (7.8). It is the original form of SVM definition.

Equation (7.16) can be expressed differently as given in Equation (7.17) which provides easier mathematical expression that can be solved. This new expression is required due to the difficulty in linear programming. In linear programming, the first problem and the second problem are complementary, which means that a solution to either one determines a solution for the both.

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j (x_i \cdot x_j) - \sum_{j=1}^{l}\alpha_j, \\ s.t. \quad & \sum_{i=1}^{l} y_i \alpha_i = 0, \end{aligned} \tag{7.17}$$

Compared with Equation (7.16) and (7.17), $w*$ and $b*$ are replaced by α. So the calculation of α replaces the previous calculations for solving equations about $w*$ and $b*$. What is more, the constraint conditions become simpler. Obviously, the Equation (7.17) is similar to Equation (7.9), which actually introduced the Lagrange Function (Equation 7.18):

$$L = f(x) - \alpha c(x); \tag{7.18}$$

$f(x)$ and $c(x)$ in Equation (7.16) is Equation (7.19),

$$\begin{cases} \min f(x) = \frac{1}{2}\|w\|^2 \\ c(x) = y_i((w \cdot x_i) + b) \geq 1, i = 1, ..., l \end{cases} \tag{7.19}$$

So

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l}\alpha_i (y_i((w \cdot x_i) + b) - 1), \tag{7.20}$$

where $\alpha = (\alpha_1, ..., \alpha_l)^T \in R_+^l$.

In optimisation field, the minimum of Equation (7.20) can be transferred to solve Equation (7.21).

$$\begin{cases} \nabla_b L(w,b,\alpha) = 0 \\ \nabla_w L(w,b,\alpha) = 0 \end{cases} \tag{7.21}$$

So, Equation (7.22) and (7.23) can be obtained as:

$$\begin{cases} \sum_{i=1}^{l} y_i \alpha_i = 0, & (7.22) \\ w - \sum_{i=1}^{l} \alpha_i y_i x_i = 0, & (7.23) \end{cases}$$

Introduce Equation (7.22) and (7.23) into Equation (7.20).

$$\begin{aligned} L(w,b,\alpha) &= \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} \alpha_i (y_i((w \cdot x_i) + b) - 1) \\ &= \frac{1}{2}\left\|\sum_{j=1}^{l} \alpha_j y_j x_j\right\|^2 - \sum_{i=1}^{l} \alpha_i (y_i(((\sum_{j=1}^{l} \alpha_j y_j x_j) \cdot x_i) + b) - 1) \\ &= \frac{1}{2}\left\|\sum_{j=1}^{l} \alpha_j y_j x_j\right\|^2 - (\sum_{i=1}^{l} \alpha_i y_i \sum_{j=1}^{l} \alpha_j y_j x_j \cdot x_i + \sum_{i=1}^{l} \alpha_i y_i b - \sum_{i=1}^{l} \alpha_i) \qquad (7.24) \\ &= \frac{1}{2} \times 0^2 - (\sum_{i=1}^{l} \alpha_i y_i \sum_{j=1}^{l} \alpha_j y_j x_j \cdot x_i + 0 \times b - \sum_{i=1}^{l} \alpha_i) \\ &= -\sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i y_i \alpha_j y_j x_j \cdot x_i + \sum_{i=1}^{l} \alpha_i \end{aligned}$$

Now, come to Equation (7.18). According to Equation (7.19), $c(x) \geq 0$ and $\alpha = (\alpha_1, ..., \alpha_l)^T \in R_+^l$.

So $\alpha c(x) \geq 0$.

Assume that

$$\max_{\alpha \in R^+} L = \max_{\alpha \in R^+} (f(x) - ac(x)).$$

When $c(x) = 0$, L will obtain the maximum. In this time, $\max_{\alpha \in R^+} L = \max_{\alpha \in R^+} (f(x) - 0 \times \alpha) = f(x).$

So $\min f(x)$ can be transferred to $\min \max L$.

$$\min_{x \in \chi} f(x) = \min_{x \in \chi} (\max_{\alpha \in R^+} L);$$

Therefore, introduced Equation (7.24) into Equation (7.19), Equation (7.25) can be obtained.

$$\max_{\alpha \in R^+} L = \max_{\alpha \in R^+} (-\sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i y_i \alpha_j y_j x_j \cdot x_i + \sum_{i=1}^{l} \alpha_i).$$

$$\min_{x\in\chi}\max_{\alpha\in R^+} L = \min_{x\in\chi}\max_{\alpha\in R^+}(-\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i y_i \alpha_j y_j x_j \cdot x_i + \sum_{i=1}^{l}\alpha_i)$$
$$= \min_{x\in\chi}\min_{\alpha\in R^+}(\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i y_i \alpha_j y_j x_j \cdot x_i - \sum_{i=1}^{l}\alpha_i) \tag{7.25}$$

So the second problem can be constructed as Equation (7.26).

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j (x_i \cdot x_j) - \sum_{j=1}^{l}\alpha_j, \\ s.t. \quad & \sum_{i=1}^{l} y_i \alpha_i = 0, \\ & \alpha_{\mathrm{i}} \ge 0, i = 1,...,l, \end{aligned} \tag{7.26}$$

So a new stylization procedure to linear problem, which is solving the second problem of stylization procedure (b), can be shown as follows. (Deng and Tian 2004)

<u>Stylization procedure (b):</u>

(1) Assume that the sample is

$T = \{(x_1, y_1),...,(x_l, y_l)\} \in (X \times Y)^l$, $\mathrm{x_i} \in \mathrm{X} = R^n, y_i \in Y = \{1,-1\}, i = 1,...,l;$

(2) Construct and solve (7.26)

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j (x_i \cdot x_j) - \sum_{j=1}^{l}\alpha_j, \\ s.t. \quad & \sum_{i=1}^{l} y_i \alpha_i = 0, \\ & \alpha_{\mathrm{i}} \ge 0, i = 1,...,l, \end{aligned}$$

and obtain the solution $\alpha^* = (\alpha_1^*,...,\alpha_l^*)^T$.

(3) Calculate $w^* = \sum_{i=1}^{l} y_i \alpha_i^* x_i$; and select positive component α_j^* of α^* and calculate the $b^* = y_j - \sum_{i=1}^{l} y_i \alpha_i^* (x_i \cdot x_j);$

(4) Construct hyper plane $(w^* \cdot x) + b^* = 0;$ and obtain decision function:

$$f(x) = \text{sgn}((w^* \cdot x) + b^*), \text{ or } f(x) = \text{sgn}(\sum_{i=1}^{l} \alpha_i^* y_i (x_i \cdot x) + b^*),$$

Correspondingly, a stylization procedure for nonlinearly separable problem can be shown as follows (Deng and Tian 2004).

<u>Stylization procedure (c):</u>

(1) Assume that the sample is

$T = \{(x_1, y_1), ..., (x_l, y_l)\} \in (X \times Y)^l, \ \ \mathrm{x_i} \in \mathrm{X} = R^n, y_i \in Y = \{1, -1\}, i = 1, ..., l;$

(2) Select penalty parameter C, then construct and solve (7.27)

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j (x_i \cdot x_j) - \sum_{j=1}^{l} \alpha_j, \\ s.t. \quad & \sum_{i=1}^{l} y_i \alpha_i = 0, \\ & 0 \le \alpha_{\mathrm{i}} \le C, i = 1, ..., l, \end{aligned} \tag{7.27}$$

and obtain the solution $\alpha^* = (\alpha_1^*, ..., \alpha_l^*)^T$.

(3) Calculate $w^* = \sum_{i=1}^{l} y_i \alpha_i^* x_i;$ and select positive component $0 < \alpha_j^* < C$ of α^* and calculate the $b^* = y_j - \sum_{i=1}^{l} y_i \alpha_i^* (x_i \cdot x_j);$

(4) Construct hyper plane $(w^* \cdot x) + b^* = 0;$ and obtain decision function:

$$f(x) = \text{sgn}((w^* \cdot x) + b^*), \text{ or } f(x) = \text{sgn}(\sum_{i=1}^{l} \alpha_i^* y_i (x_i \cdot x) + b^*),$$

In comparison to stylization procedure (b), stylization procedure (c) adds the penalty parameter C. For nonlinearly separable problem, a margin can not always be positive. Therefore if one still wants to use the hyper plane to distinguish the data, a slack variable $\xi_i \geq 0, i = 1,...,l;$ is introduced. Equation (7.16 b) can be transferred to Equation (7.28).

$$s.t. \quad \mathrm{y_i}((w \bullet x_i) + b) \geq 1 - \xi_i, \quad i = 1,...,l \tag{7.28}$$

Obviously, if ξ_i is big enough, there is always a hyper plane that can be found. But, if ξ_i is too big, it is very hard to control. So a penalty parameter C is introduced. In this case, Equation (7.18) can be transferred to Equation (7.29).

$$\begin{aligned}
&\min \quad \Phi(w) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i, && \text{(a)} \\
&s.t. \quad \mathrm{y_i}((wgx_i) + b) \geq 1 - \xi_i, \quad i = 1,...,l, && \text{(b)} \\
&\qquad \xi_i \geq 0, i = 1,..., j,
\end{aligned} \tag{7.29}$$

$$L(w,b,\alpha,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i - \sum_{i=1}^{l}\alpha_i(\mathrm{y_i}((wgx_i) + b) - 1 + \xi_i)$$

$$\nabla_w L(w,b,\alpha,\xi) = w - \sum_{i=1}^{l}\alpha_i \mathrm{y_i} x_i = 0$$

$$\nabla_b L(w,b,\alpha) = \sum_{i=1}^{l}\alpha_i \mathrm{y_i} = 0$$

$$\nabla_\xi L(w,b,\alpha) = C - \sum_{i=1}^{l}\alpha_i = 0$$

$$\begin{aligned}
L(w,b,\alpha) &= \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i - \sum_{i=1}^{l}\alpha_i(\mathrm{y_i}((wgx_i) + b) - 1 + \xi_i) \\
&= \frac{1}{2}(\sum_{i=1}^{l}\alpha \mathrm{y_i} x_i)^2 + C\sum_{i=1}^{l}\xi_i - \sum_{i=1}^{l}\alpha_i(\mathrm{y_i}((\sum_{j}\alpha_j \mathrm{y}_j x_j gx_i) + b) - 1 + \xi_i) \\
&= 0 + C\sum_{i=1}^{l}\xi_i - (\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i \mathrm{y_i}\alpha_j \mathrm{y}_j x_j x_i + \sum_{i=1}^{l}\alpha_i \mathrm{y_i} b - \sum_{i=1}^{l}\alpha_i + \sum_{i=1}^{l}\alpha_i\xi_i) \\
&= C\sum_{i=1}^{l}\xi_i - \sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i \mathrm{y_i}\alpha_j \mathrm{y}_j x_j x_i - 0 + \sum_{i=1}^{l}\alpha_i - \sum_{i=1}^{l}\alpha_i\xi_i) \\
&= -\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i \mathrm{y_i}\alpha_j \mathrm{y}_j x_j x_i + \sum_{i=1}^{l}\alpha_i
\end{aligned}$$

So, the second problem of Equation (7.29) can be constructed as Equation (7.27).

Thus, they can be extended to nonlinear problem. For nonlinear problem, a hyper plane can not be used, instead hyper surface is needed. The problem of seeking hyper surface (3D) can be transferred to seeking hyper plane (2D) via dimension changing.

Stylization procedure (c) is a good procedure for understanding, but for particle operation, it is hard because there is no selection standard for parameter C. For better operation, an improved procedure is introduced. (Deng and Tian 2004)

Stylization procedure (d):

(1) Assume that the sample is

$T = \{(x_1, y_1), \dots, (x_l, y_l)\} \in (X \times Y)^l$, $x_i \in X = \mathbb{R}^n, y_i \in Y = \{1, -1\}, i = 1, \dots, l;$

(2) Select proper kernel function $K(x, x')$ and parameter ν, then construct and solve (7.30)

$$
\begin{aligned}
\min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j K(x_i, x_j), \\
s.t. \quad & \sum_{i=1}^{l} y_i \alpha_i = 0, \\
& 0 \le \alpha_i \le \frac{1}{l}, i = 1, \dots, l, \\
& \sum_{i=1}^{l} \alpha_i \ge \upsilon,
\end{aligned}
\tag{7.30}
$$

and obtain the solution $\alpha^* = (\alpha_1^*, \dots, \alpha_l^*)^T$.

(3) Select $j \in S_+ = \{i \,|\, \alpha_i^* \in (0, 1/l), y_i = 1\}, k \in S_- = \{i \,|\, \alpha_i^* \in (0, 1/l), y_i = -1\}$ and calculate the $b^* = -\frac{1}{2} \sum_{i=1}^{l} y_i \alpha_i^* (K(x_i, x_j) + K(x_i, x_k));$

(4) Construct decision function:

$$f(x) = \mathrm{sgn}\left(\sum_{i=1}^{l} \alpha_i^* y_i K(x_i, x) + b^*\right),$$

Here, parameter ν replaces the parameter C.

If $\rho^* > 0$, then：

If number of margin error samples (shown in Figure 7.5) is p, then $\nu \geq p / l$;

If number of support vectors is q, then $\nu \leq q / l$;

ρ^* is the solution of follows:

$$
\begin{aligned}
&\min \quad T(w,\xi,\rho) = \tfrac{1}{2}\|w\|^2 - \nu\upsilon + \frac{1}{l}\sum_{i=1}^{l}\xi_i, && \text{(a)} \\
&s.t. \quad \mathrm{y_i}((w g x_i) + b) \geq \rho - \xi_i, \quad i = 1,\ldots,l, && \text{(b)} \qquad (7.31) \\
&\qquad\ \ \xi_i \geq 0, i = 1,\ldots, j, \rho \geq 0 &&
\end{aligned}
$$

ξ^* is the margin error samples.

In this section, the whole framework of SVM application from Stylization procedure (d) has been built. In the next section, the detailed case study will be presented. There are two virtual parts within learning function which need to be created: the specialist committee and the technology manager. The specialists committee gives the evaluation to the case and the manager allots the weightings to the specialists. Both factors are critical for this FMADM method. In the following section, these two virtual parts will be independently built and then put together.

7.5 Case Study of LMFMADM in the ship stability design

Before starting to build the virtual specialist committee, there are several principles, which must be clearly defined in order to ensure the items to be processed smoothly. The first one is the structure of virtual specialist committee with the learning function. The structure should be simple and clear to both the designers who will use the results and the trainers who will train the virtual specialist committee before it is put into use formally. Especially for the designers, they may not be expert for decision making but just they need to know how to use the software. So the specialist committee should be easily operated without additional decision making knowledge. The second principle is that the virtual specialist committee should have the ability to handle the linguistic properties. In other words, after the linguistic attribute is transferred to the numerical attribute, the virtual specialist committee can control both discrete and continuous numbers. The last one is that the training work before use should be kept on renewing the knowledge and reforming the context. This requires that the continuing output should be introduced as a training set input after revised by the designers. This action will expand the training set and improve the accuracy and stability of the system.

The SVM is selected as machine learning approach to build the virtual specialist and technology manager while the JAVA language is employed for coding.

In order to create a successful method, the first important thing is to build the model which includes selection of the training sets, the detailed SVM approaches and the detailed parameters setting.

In this model, every specialist will provide the evaluation for every attribute of the alternative designs. So before the system can give the prediction, the training set containing these evolutions should be given to SVM.

The training set is very important for the application because the prediction is given based on them. Even if the training sets have very small errors, the final prediction will probably have a large deviation. So the training set should be checked carefully before it is utilised by the designers.

The reason that SVM can well control the classification for the small sample is that the SVM is seeking the minimum of SRM according to Equation (7.3) and (7.4). The over-fitting of ANN caused by the machine learning is too complex. The machine learning ensures that the $R_{emp}[f]$ is very small but VC dimension is very big, which means the expected risk is still very high. The under-fitting of ANN caused by the machine learning is too simple. In this situation, the $R_{emp}[f]$ is very high but VC dimension is very small. So the expected risk is still very high. The SVM is seeking for the structure risk minimisation (SRM) and can avoid these problems.

7.5.1 Introduction of Case Study of LMFMADM in the ship stability design

The case study is the same as previous work (Ölçer, Tuzcu et al. 2005)，which is used to demonstrate the learning function of FMADM method. In previous case, the specialists in different areas are convened to make an evaluation of every design solution. Then the weights are allocated for the ranking and analysis, which are processed to obtain the final design.

In previous work, the authors wanted to select one design as the final design from six Pareto-optimal design alternatives (PODAs). Three experts from three subjects including production engineer, designer and operator gave evaluation from six attributes. Because the proposed paper did not provide the values of six PODAs, the solutions in another optimisation case study (Cui and Turan 2009) will be used to replace the solutions of original work.

Therefore, the aim of this case study is to build the virtual experts and technology manager via LMFMADM to evaluate the PODAs from optimisation on stability design. There are six designs which will be evaluated by the human experts according to six attributes and then these data will be analysed via LMFMADM. The ten designs randomly selected from PODAs will evaluate the ability of this method.

In this study, the situation is that there are no specialists and corresponding weightings in the design but only previous design experience, which also is the common situation in design process. The SVM will be applied to construct the virtual specialist committee to provide the function of specialists and of technology manager.

7.5.2 Parameters setting of Case Study of LMFMADM

In this section, A represents the attributes of design solutions which are expressed as X. The specialists are E. There are six attributes (A_1 to A_6), six solutions (X_1 to X_6) and three specialists (E_1 to E_3).

There are three objectives attributes (A_1, A_2 and A_3) and three subjective attributes, which are listed below:

A_1 : cargo capacity

A_2 : Hs value

A_3 : KG limiting value

A_4 : producbility

A_5 : ease of maintenance and repair

A_6 : loading/unloading efficiency of arrangement

These attributes are taken from previous work (2005). The first three attributes are numerical attributes, which obtained from the optimisation algorithm and they do not need to evaluate. The last three attributes are attributes and they need experts to make an evaluation based on the first three attributes.

The six solutions (X_1 to X_6) are shown in Table 7.1:

Table 7. 1 The optima solutions for training in SVM

	X_1	X_2	X_3	X_4	X_5	X_6
A_1 (Hs)	5.14501m	5.13495m	5.12806m	5.13548m	5.00481m	4.9279m
A_2 (KG Lim)	14.7361m	14.7551m	14.8708m	14.7551m	14.9139m	14.9827m
A_3 (Cap.)	12 lines	10 lines	10 lines	10 lines	8 lines	12 lines

From Table 7.1, X_1 to X_6 are six solutions taken from previous work (Cui and Turan 2009). All of these solutions are Pareto optimisations and are selected randomly. These data will be analysed by the human experts as the training data. The rules and regulations found by analysis will be used to as the classifier of the oncoming designs. In this case study, other solutions will be used to check the classifier.

The group consists of three experts:

E_1 production engineer

E_2 designer

E_3 operator

The experts are selected according to previous work (Ölçer, Tuzcu et al. 2006) and different expert assesses different aspect of solutions. For production engineer (E_1), this expert pays attention to the building cost. The designer expert (E2) will focus on the design performance when the operator (E3) will assess the easy operation. The evaluations of three subjective attributes of specialists are displayed in Table 7.2.

Table 7. 2 Experts' evaluation of six training PODAs under three subjective attributes and their corresponding fuzzy numbers

		X_1			X_2			X_3		
		E_1	E_2	E_3	E_1	E_2	E_3	E_1	E_2	E_3
A_4	Expert opinions	Good	Good	Good	Good	Good	Good	Good	Bad	Bad
	Fuzzy numbers									
A_5	Expert opinions	Good	Bad	Bad	Good	Bad	Bad	Bad	Good	Good
	Fuzzy numbers									
A_6	Expert opinions	Good	Good	Good	Good	Good	Good	Bad	Good	Good
	Fuzzy numbers									

		X_4			X_5			X_6		
		E_1	E_2	E_3	E_1	E_2	E_3	E_1	E_2	E_3
A_4	Expert opinions	Good	Good	Good	Bad	Bad	Bad	Good	Bad	Good
	Fuzzy numbers									
A_5	Expert opinions	Good	Good	Good	Bad	Good	Good	Bad	Good	Good
	Fuzzy numbers									
A_6	Expert opinions	Good	Good	Good	Bad	Bad	Good	Good	Good	Bad
	Fuzzy numbers									

Table 7. 3 Attributes' properties and weightings of attributes and experts

Attributes	type of assessment	type of attribute	Relative importance	w	E_1		E_2		E_3	
					R.I.	we_1	R.I.	we_2	R.I.	we_3
A_1 (the most important)	Crisp	Objective	100	0.22						
A_2 (the moderate important)	Crisp	Objective	75	0.17						
A_3 (the most important)	Crisp	Objective	100	0.22						
A_4 (the least important)	Linguisitic	Subjective	50	0.11	1	0.4	1	0.4	0.5	0.2
A_5 (the least important)	Linguisitic	Subjective	50	0.11	0.3	0.15	0.7	0.35	1	0.5
A_6 (the moderate important)	Linguisitic	Subjective	75	0.17	0.4	0.2	0.6	0.3	1	0.5

The training set should be considered carefully to satisfy the following principles:

✓ The training set should have enough examples. Although the SVM is suitable for the small training set, the training set should contain as many examples as possible.

- ✓ The attributes in the training set should be appropriate because they will affect the results of training.

For easy understanding, the experts' evaluation of six training PODAs is limited to two results: good and bad. The evaluation results are shown in Table 7.2. The problem is transferred to build a model which can intelligently give an evaluation for new design and the evaluation is classified into two levels: Good or Bad. Table 7.3 shows

In the second step, the designers should select appropriate approaches of SVM including core functions and parameters. For this problem, different approaches of SVM are selected to compare the efficiency.

In order to simplify the objective function and reduce the complexity, the problem is divided into smaller problems. The evaluation results of the specialists are shown in Table 7.2. In order to further clearly express the relationship between the expert and attribute, Table 7.4 shows all the experts' evaluation to the attribute A4. From Table 7.4, it can be easily seen that experts' assessments are different for the same attribute of every PODA. The expert E1 thinks the PODA x_5 is good when others consider it is bad from the view of their subject. The LMFMADM should learn the knowledge from these experts and apply this knowledge to make an evaluation in the next application.

Table 7. 4 Experts' evaluation of six training PODAs for attribute A4

Expert E1	Expert E2	Expert E3
Good=1	Good=1	Good=1
x_1,x_2,x_3,x_4,x_6	x_1,x_2,x_4	x_1,x_2,x_4
Bad=-1	Bad=-1	Bad=-1
x_5	x_3,x_5,x_6	x_3,x_5,x_6

7.5.3 Virtual experts

The construction of virtual experts is complex task. Different design requirement needs different experts and even the same design cases but slight changes will affect the robustness of experts' evaluation.

Because the expert comes from different area and has different preference, the expert group should be divided into individual for better moulding of experts' personalities. Firstly, the specialist E1 needs to be constructed, and for attribute A4, the evaluation of E1 includes two sets:

$$\begin{aligned}&G_{E1A4}=\{((5.14501,14.7361,12),1),((5.13495,14.7551,10),1),\\&((5.12806,14.8708,10),1),(5.13548,14.7551,10),1),((4.9279,14.9827,12),1)\}\end{aligned} \quad (7.32)$$

$$B_{E1A4}=\{((5.00481,14.9139,8),-1)\};$$

Equation (7.32) shows that two sets are created: G and B. A is the set of attributes which expert E_1 thinks 'good'. The G_{E1A4} includes five PODAs and the numerical value '1' is given to represent that this category is good. Correspondingly, set B is the PODA aggregate of 'bad' and the numerical value '-1' is given as representation.

Now application of stylization procedure (d) into practical design can be considered. From Table 7.1 and Table 7.2, the training sample set of specialist E_1 for attribute A4 can be constructed as shown in Table 7.5:

Table 7. 5 The training sample set of specialist E_1 for attribute A4

Train samples	X_1		X_2		X_3		X_4		X_5		X_6	
A_1 (Hs)	5.14501	m	5.13495	m	5.12806	m	5.13548	m	5.00481	m	4.9279	m
A_2 (KG Lim)	14.7361	m	14.7551	m	14.8708	m	14.7551	m	14.9139	m	14.9827	m
A_3 (Cap.)	12	Lines	10	Lines	10	Lines	10	Lines	8	Lines	12	Lines
A_4(y)	1		1		1		1		-1		1	
Category	G1		G2		G3		G4		B1		G5	

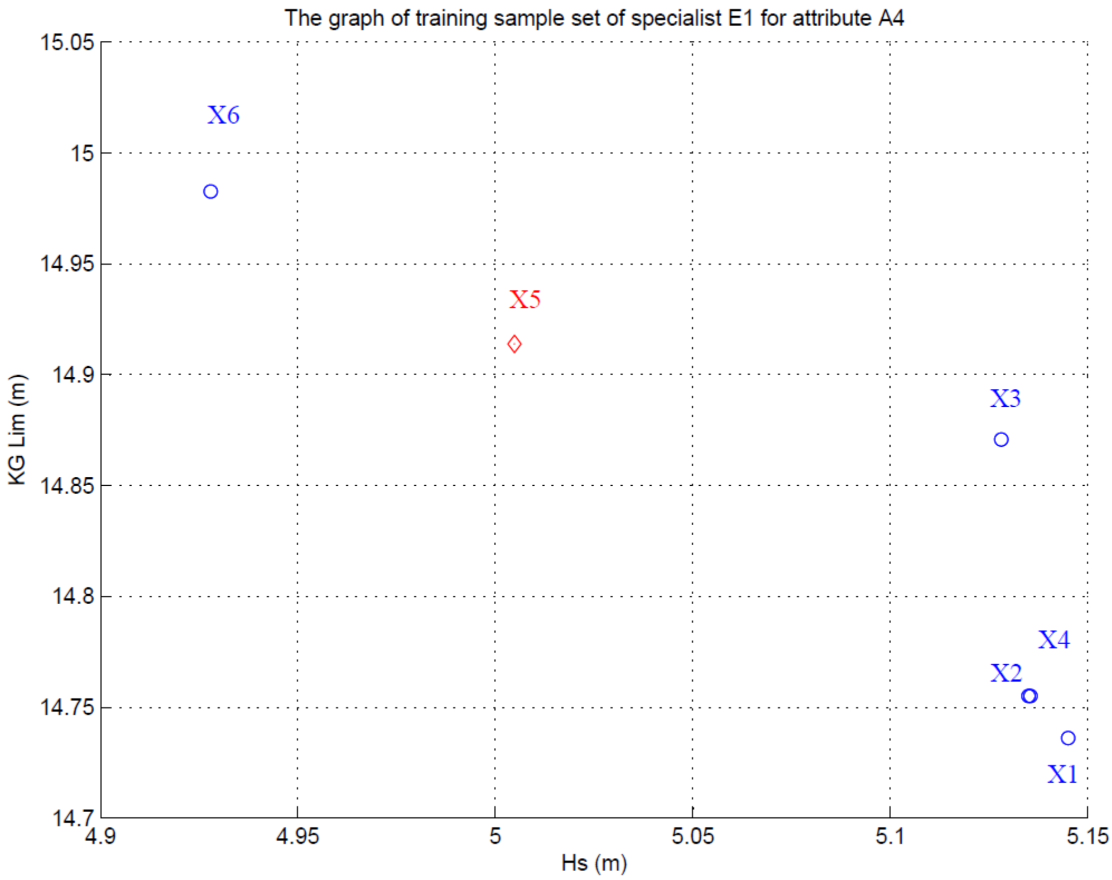


Figure 7. 7 The graph of training sample set of specialist E_1 for attribute A4

For a better display of the run results, a 2D graph is given, which means only A1 and A2 in Table 7.5 are selected in the first step. The Figure 7.7 shows this 2D graph of the training sample set of specialist E_1 for attribute A4. The blue represents the category G (good) and red means the category B (Baddeley).

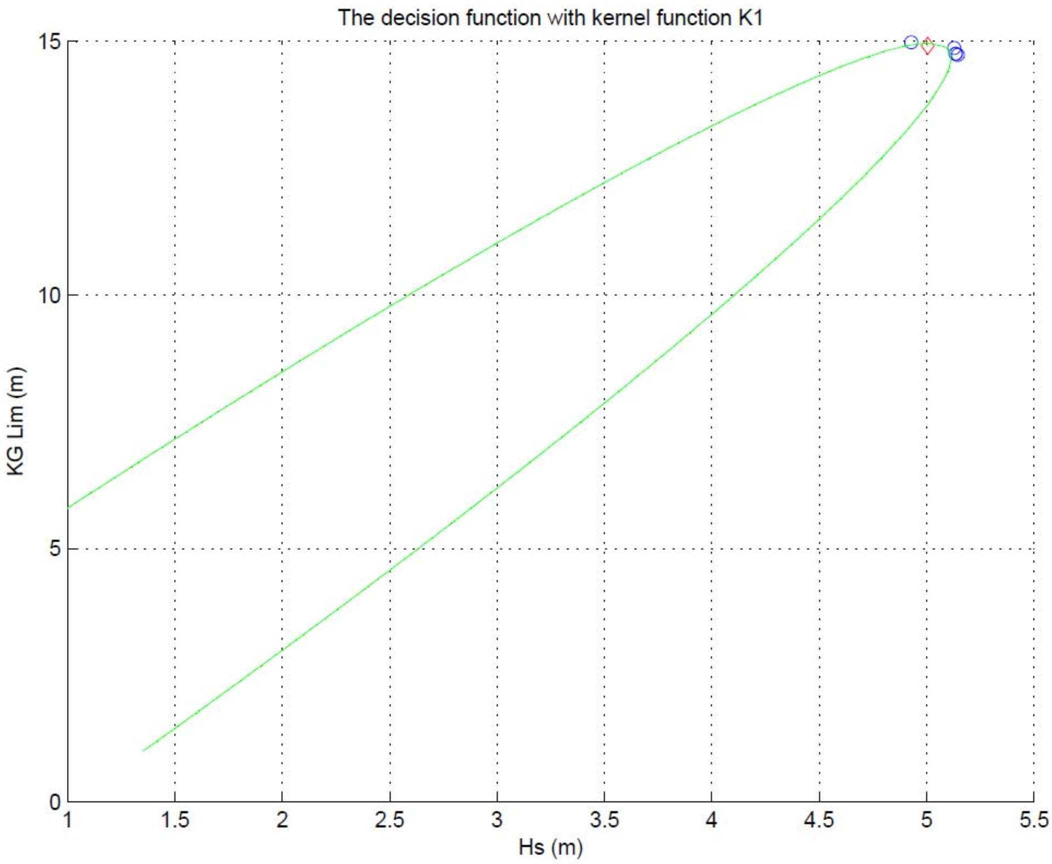
In the next steps, an appropriate kernel function should be selected to promote the calculation. Up to now, there is not a general method to select kernel functions. In this case, several popular kernel functions will be tested to check the effectiveness. In the following, the popular kernel functions will be tested.

First, kernel function is a linear function: K=$(x_i \cdot x_j)$.

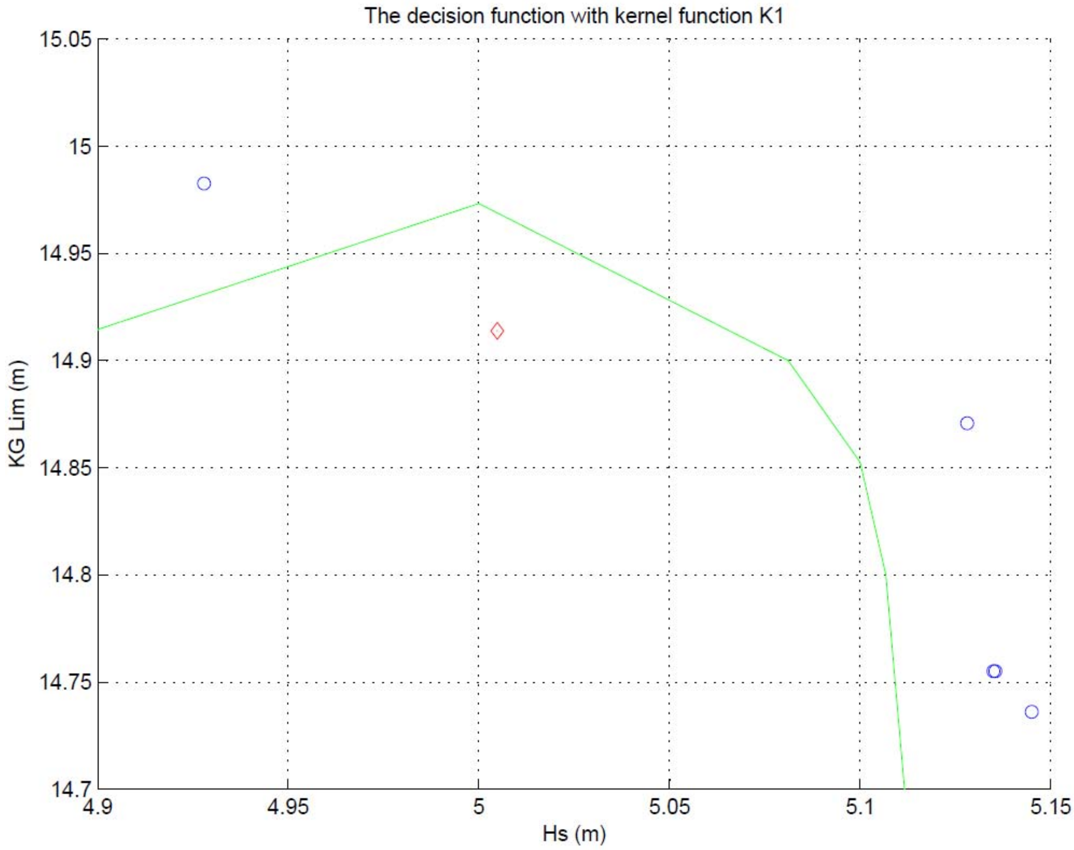
Because Table 7.5 is nonlinear data, the linear kernel function can not find the solution even if C tends to go to infinity.

Second, kernel function is Polynomial (homogeneous) function: K1=$(x_i \cdot x_j)^2$;

$$f_{K1} = \text{sgn}(\sum_{i=1}^{l} y_i \alpha_i K_1(x_i, x)\text{-}123.9199);$$



(a)

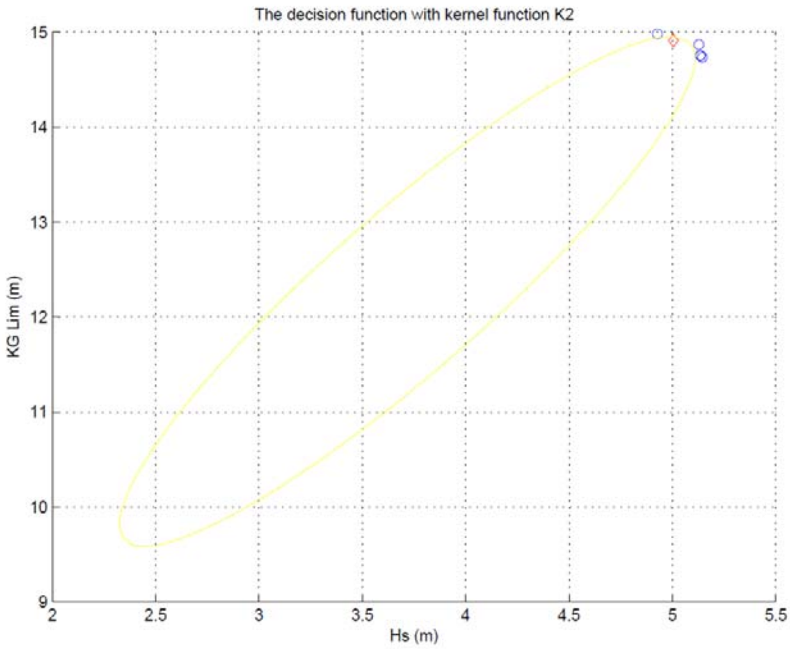


(b)

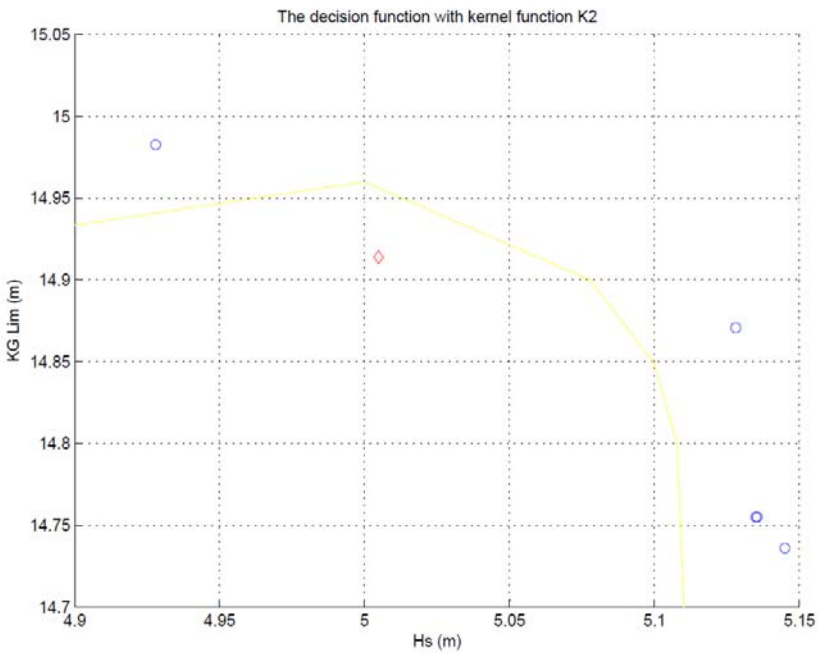
Figure 7. 8 The decision function of K1=$(x_i \cdot x_j)^2$ on training sample set of specialist E_1 for attribute A4

Third, kernel function is Polynomial (inhomogeneous) function: K2=$((x_i \cdot x_j)+100)^3$;

$$f_{K2} = \text{sgn}(\sum_{i=1}^{l} y_i \alpha_i K_2(x_i, x) + 728.2246);$$



(a)



(b)

Figure 7. 9 The decision function of K2= $((x_i \cdot x_j)+100)^3$ on training sample set of specialist E_1 for attribute A4

The last one, kernel function is Gaussian Radial basis function: K3= $\exp(-\frac{\|xi-xj\|^2}{2\sigma^2})$;

$$f_{K3} = \mathrm{sgn}(\sum_{i=1}^{l} y_i \alpha_i K_3(x_i, x) + 73.7188);$$

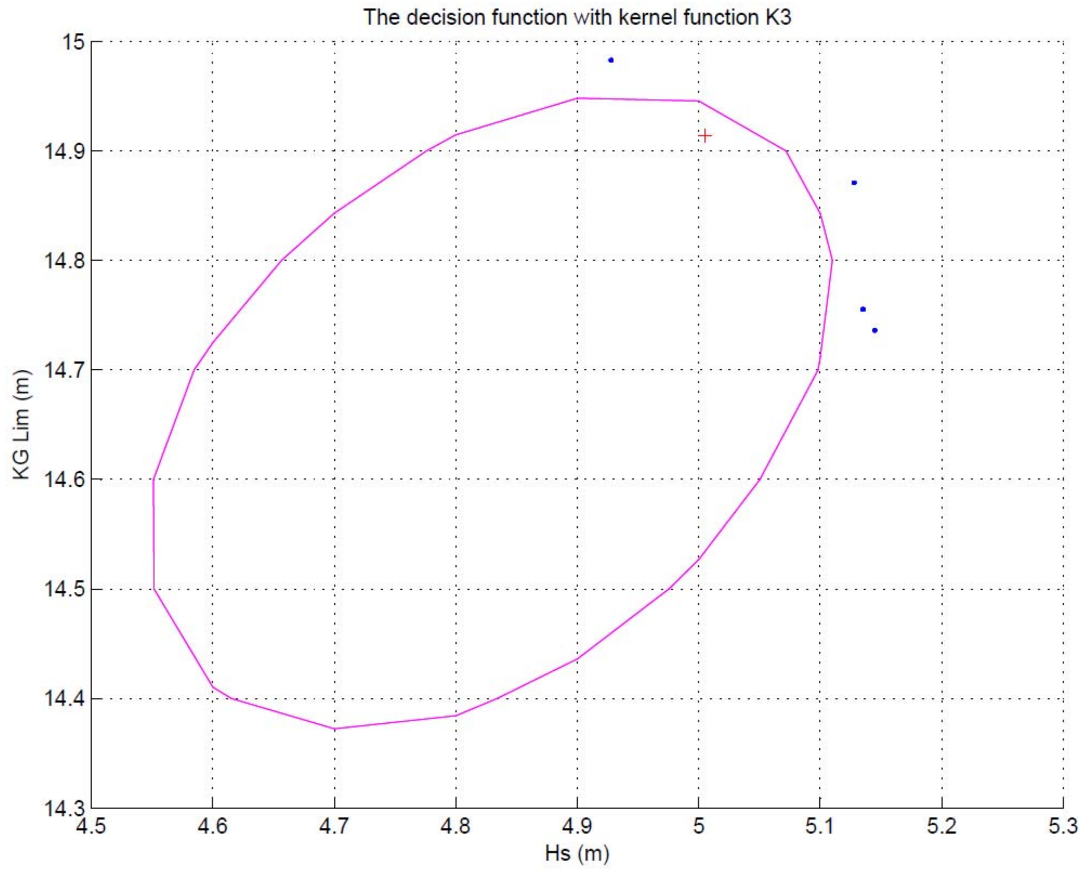
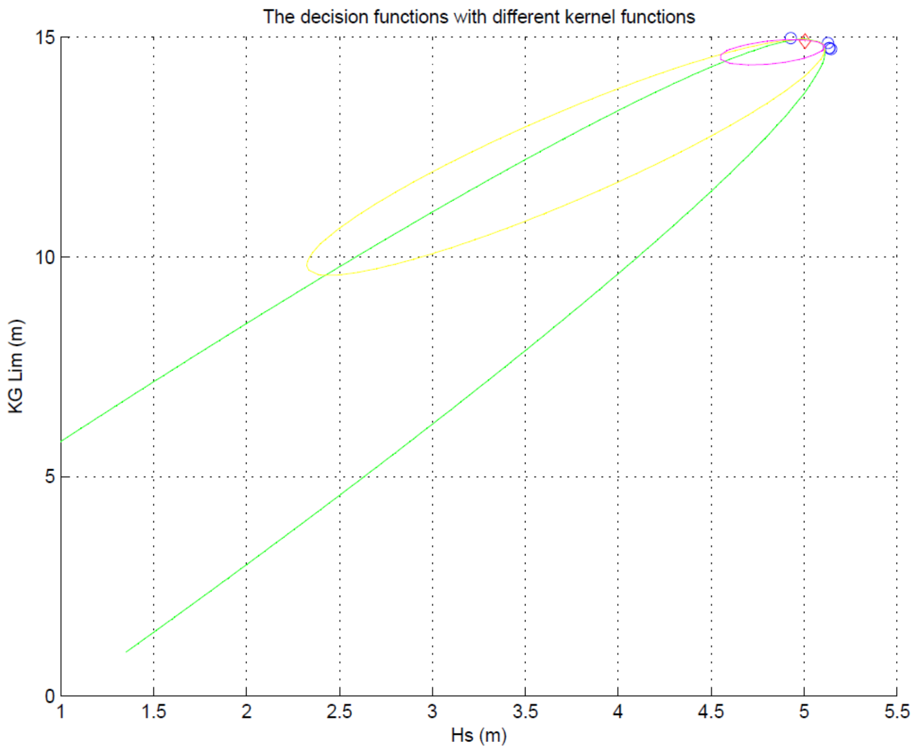


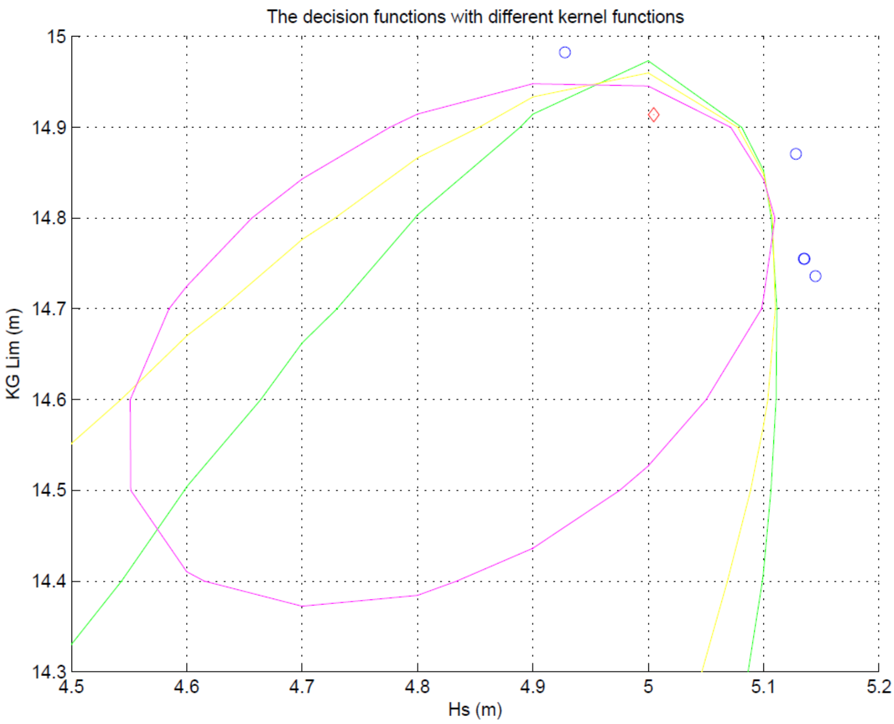
Figure 7. 10 The decision function of K3= $\exp(-\frac{\|xi - xj\|^2}{2\sigma^2})$ on training sample set of specialist E_1 for attribute A4

From Figure 7.8, 7.9 and 7.10 it can be seen that, several popular kernel functions all can correctly distinguish the data. Because $K_3 = \exp(-\frac{\|xi - xj\|^2}{2\sigma^2})$ has the minimum Area shown in Figure 7.10, the kernel K_3 is selected as the best classifier for this case. This means if the new data belongs to the inner circle (purple circle) in Figure 7.10, it will be good for specialist E_1 for attribute A_4. If not, it will be bad. The new running data can continue to revise the decision function.

The comparison of decision functions of different kernel functions on training sample set of specialist E_1 for attribute A4 is shown in Figure 7.10. In Figure 7.10 (a), the green line is the kernel function K1= $(x_i \cdot x_j)^2$ and yellow line is the kernel function K2= $((x_i \cdot x_j)+100)^3$ when purple line is the third kernel function K3= $\exp(-\frac{\|xi - xj\|^2}{2\sigma^2})$. It can be seen from Figure 7.10 (a) that every line can make a partition of the training data. In proposed design support system, the kernel function K3 is used as default method. If the users of the system do not select the special kernel function, K3 is pointed to make a classification.



(a)



(b)

Figure 7. 11 Comparison of decision functions of different kernel functions on training sample set of specialist E_1 for attribute A4

For the attributes A_5 and A_6, the same method is used and the kernel function K3 is selected. The full virtual expert E1 is shown in Figure 7.12. When put the all decision function together, a systemic virtual specialists will be created.

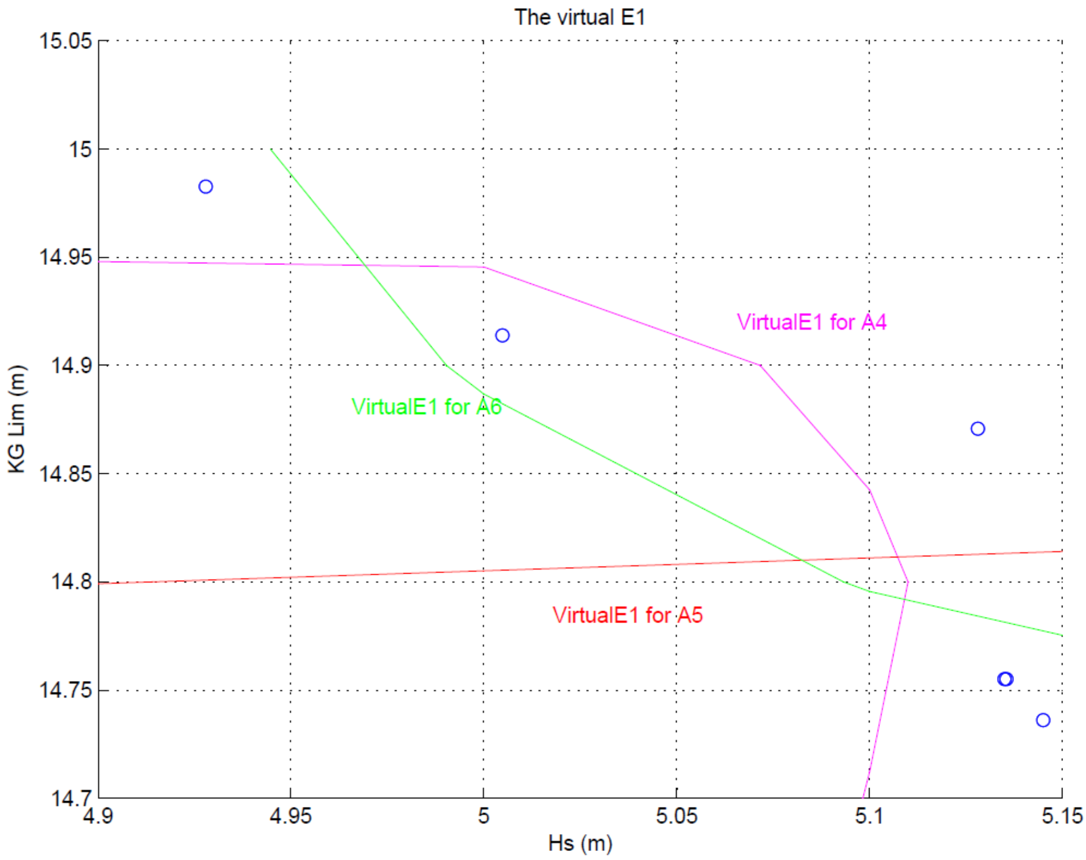


Figure 7. 12 Virtual expert E_1

After the virtual experts are created, the system will automatically use these virtual experts to make an evaluation. For better understanding, the Figure 7.13 shows the application of virtual expert E1. Suppose there are four designs which are four test points as shown in Figure 7.13 and the system needs to provide evaluation of these designs. For the attribute A4 of test point 1, virtual expert E1 will use purple line to make discrimination. The test point 1 is located above the purple line, so it belongs to category of 'good' on attribute A4. Then the evaluations are processed to Test Point 2, Test Point 3 and Test Point 4. It can be seen that all of these points are located under the purple line, so they belong to category of 'bad' on attribute A4.

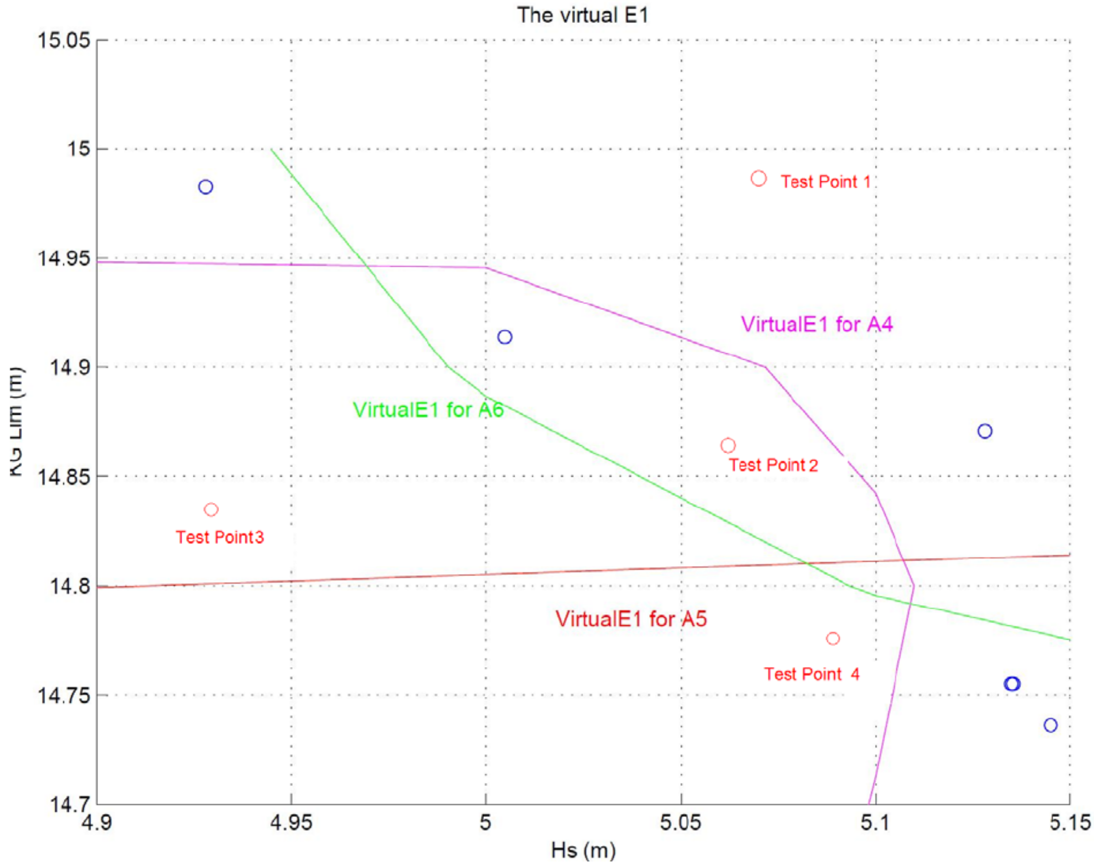


Figure 7. 13 Application of Virtual expert E_1

In the following step, the attribute A5 is taken account. The red line will be used to make the evaluations. Of the four points, the Test Point 1, Test Point 2 and Test Point 3 are above the red line. So these designs belong to category of 'good' on attribute A5. For the fourth point, Test Point 4 is below the red line, so this design belongs to category of 'bad' on attribute A5. The similar operations are processed to the evaluation of attribute A6 and the green line is used. The Test Point 1 and Test Point 2 belong to category of 'good' on attribute A6 when Test Point 3 and Test Point 4 belong to category of 'bad' on attribute A6. The final evaluations are listed in Table 7.6.

Table 7. 6 The result set of specialist E_1 for test

Test Points	Test Point 1	Test Point 2	Test Point 3	Test Point 4
A_4	Good	Bad	Bad	Bad
A_5	Good	Good	Good	Bad
A_6	Good	Good	Bad	Bad

7.6 Learning based ship design decision support system

In previous chapters, Learning Based Ship Design Support system is fully explained. In this section, the whole system will be built on the frame of multi-agent system. The method and key points of building this system, the organization of the system and the communication among agents, etc. are listed and explained. With the theory of multi-agent, which is utilized in the proposed system, there is no administer agent which manages other agents. All the agents are independently embedded into the environment as a real human society. In order to effectively organise the design actions, a smart environment is created.

The collaboration of agents is very important for the multi-agent system. A simple communion approach via smart environment has been taken into account to avoid complex dialogue between the agents. Because the multi-agent technology is not mature yet, the design decision support system will employ some simple approaches to reduce the complexity of system and keep the system focusing on the engineering application. The modularisation idea is used here to make the system more robust and feasible so that if a new technology comes, the system can rapidly replace old module with the new one.

7.6.1 Introduction of learning based design decision support system

The introduction of group technology into the ship building technology started during 1960s. In 1990s, the researchers proposed the idea of sets in automotive industry and introduced the assembly line to ship production. In ship yard management, many ship yards introduced the supply chain management from 1990s. All of these changes made a great improvement of ship production. However, the development of ship design still has no obvious improvement except for the application of CAD and CAE in the design area. The idea of decision based design and agent based design has been proposed for a long time, but it is not used in practice. The design work still follows the old spiral. One of the very important reasons is that the knowledge, which supports the ship design, is very complex and the design work has to largely depend on experience of specialists. This makes the automated design very difficult. The learning function, which is given in previous chapters, has greatly promoted the possibility to solve this problem. The other reason is that the design work was processed based on line principle by human designers. This limits the efficiency of whole ship design work. A multi-agent system is introduced in this chapter to overcome this problem.

The learning based design decision support system is an effective and intelligent tool to assist the ship designers to make a better design. It has the ability to control the whole design process via parallel design theories. Every agent is independent and has service-oriented architecture. The independence means that the agent can operate by itself based on enough information without the help from other agents. The service-oriented architecture, which is transferred from computer science, means that the agent is organized according to different service that they can provide.

The design decision support system consists of four agent groups and two interface agents. For the convenience of communication, a smart environment is created. The four agent groups are independent from each other. The smart environment is responsible for the information exchange. In other word, there is no main agent or agent group dominate other agents. The agent and agent group calculate the service function, and then send the results to the environment. The environment classifies the data and makes sure the direction to which the data are sent. This means the environment can function as the organizer of the system. But what should be kept in mind is that the multi-agent is a parallel environment and the priority does not exist in this environment. The environment just gives a route of the design work.

In the system, there is no chief agent to avoid too many tasks on one agent. If all the design work needs a chief agent to manage, the information has to wait the chief agent to respond, which will cause waste of time and delays.

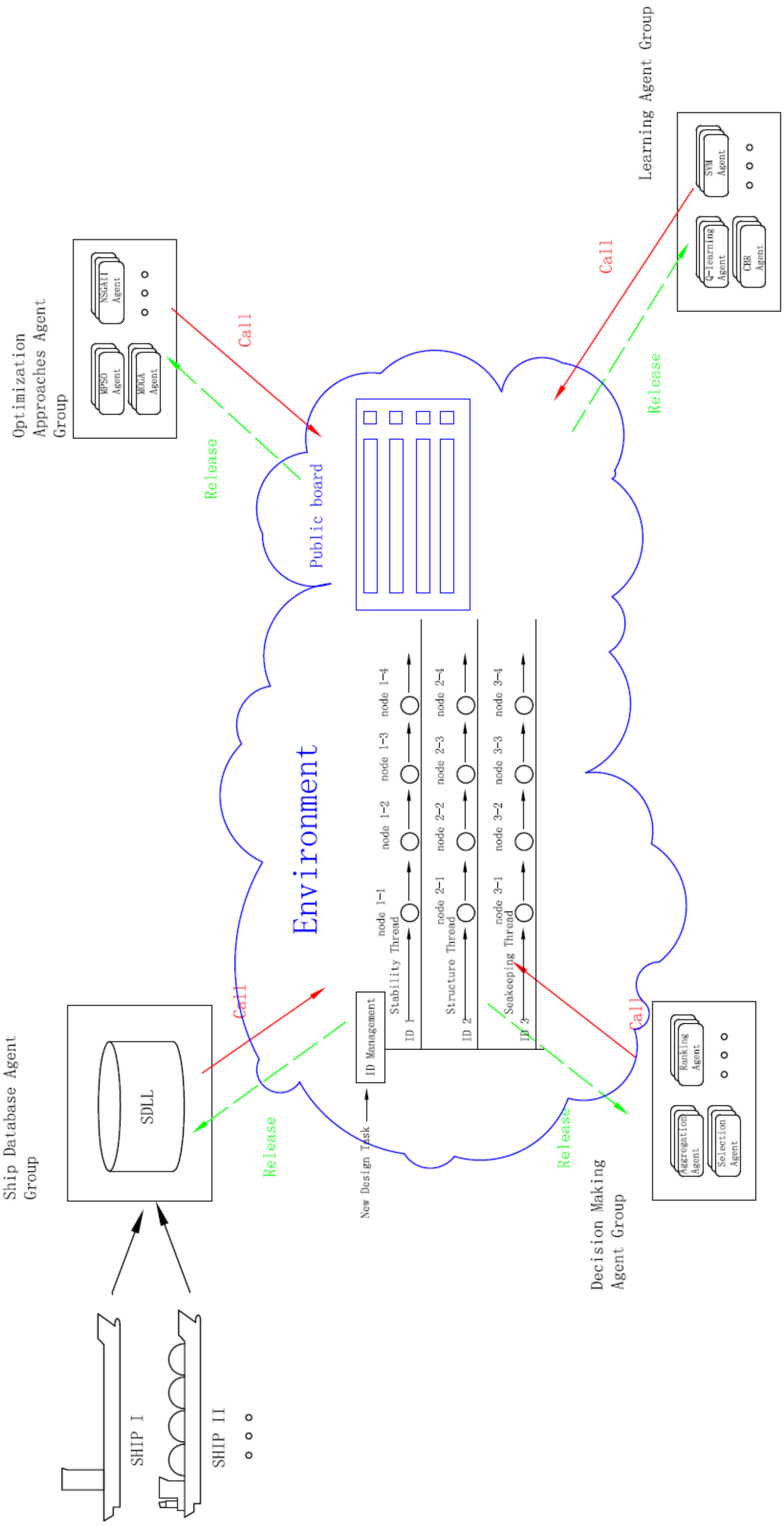


Figure 7.14 The proposed learning based design decision support system

7.6.2 Definition of agent/agent group

As mentioned before, there are four agent groups and two interface agents in the system. This section will explain the structure and function of these agent groups/agents. The important and difficult points in building and running process will also be introduced. It is noteworthy to mention that these agent groups/agents are not arranged according to the process order but based on service function. They can finish a job independently. The environment calls them according to the requirements of process. In other word, there may be many synchronizing design tasks and the environment allots the different process to different design task. The detailed process can employ the agents according to its own design requirements.

The first agent group is database agent group as shown in Figure 7.15. This agent group contains SDLL and other database used in this system. The database is the basement of the system and most of algorithms need a database to support the operation. All the databases are managed here except the temporary database, which are created in running process and will be deleted after the run. The two main databases in this group are the SDLL and decision making training database. The function of SDLL focuses on providing relationships among the design variables, the rules and regulations as constraints, reference cases etc. to assist the designers make clear about the entire design line and detailed operations. The decision making training database provides the training sets for SVM to make a prediction and support virtual specialists and the technology manager. The database also needs to be updated after runs to enlarge the training sets to improve the ability of SMV.

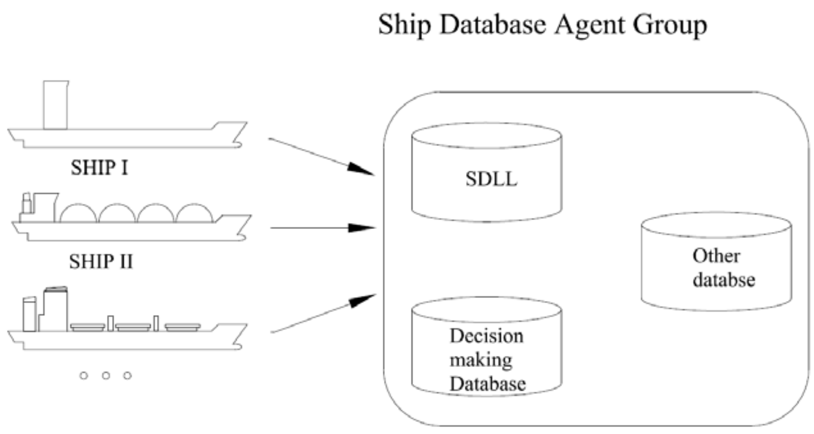


Figure 7.15 The ship database agent group

The second agent group is optimisation agent group as shown in Figure 7.16. This agent group provides the optimisation for other agent/agent group. There are four independent agents and one link agent. The two of the four independent agents are the single objective optimisation agents which are Particle swarm optimisation (PSO) agent and GA agent, and two multi-objective optimisation agents, HCPSO and NSGAII. The link agent provides the link to Mode Frontier software. So the system can use other mature optimisation approach to do the optimisation work.

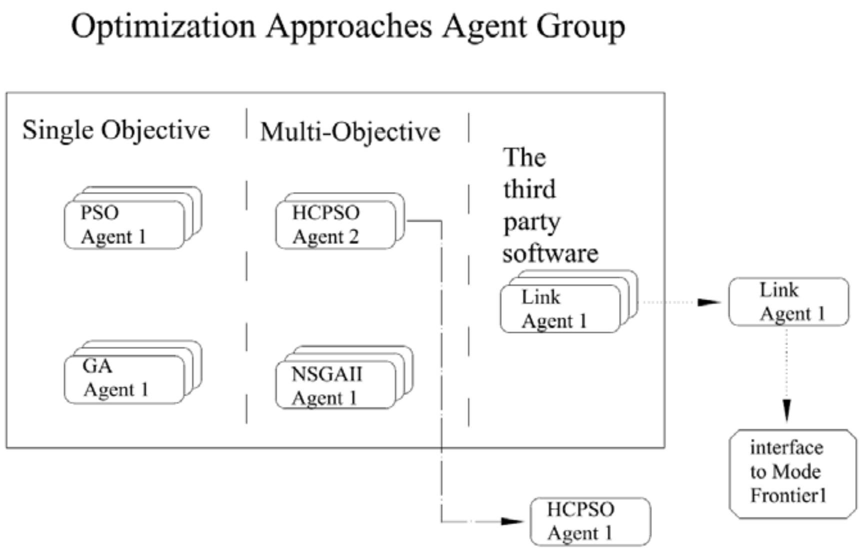


Figure 7.16 The optimisation agent group

The third agent group is learning approach agent group as shown in Figure 7.17. This agent group contains the learning approaches for operation including decision tree, CBR, Q-learning and SVM. Other agents can use the fixed learning approaches for their own function and they can also employ other learning methods freely.

Learning Approaches Agent Group

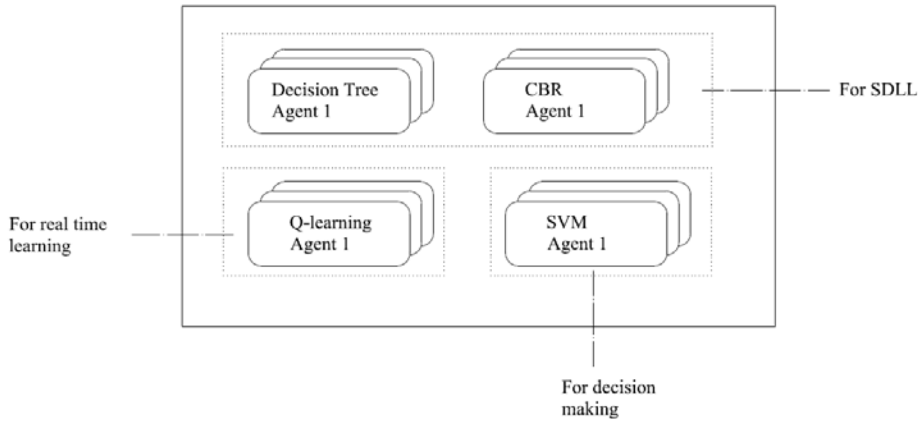


Figure 7.17 The learning approach agent group

The fourth agent group is decision making agent group as shown in Figure 7.18. This agent group contains the decision making agents for final decision. In current research, the FAMDA decision making approach together with its improved version with virtual specialist committee and technology manager can be utilized. The designers can select between these two approaches. In the future research, more decision making approaches will be introduced.

Decision Making Agent Group

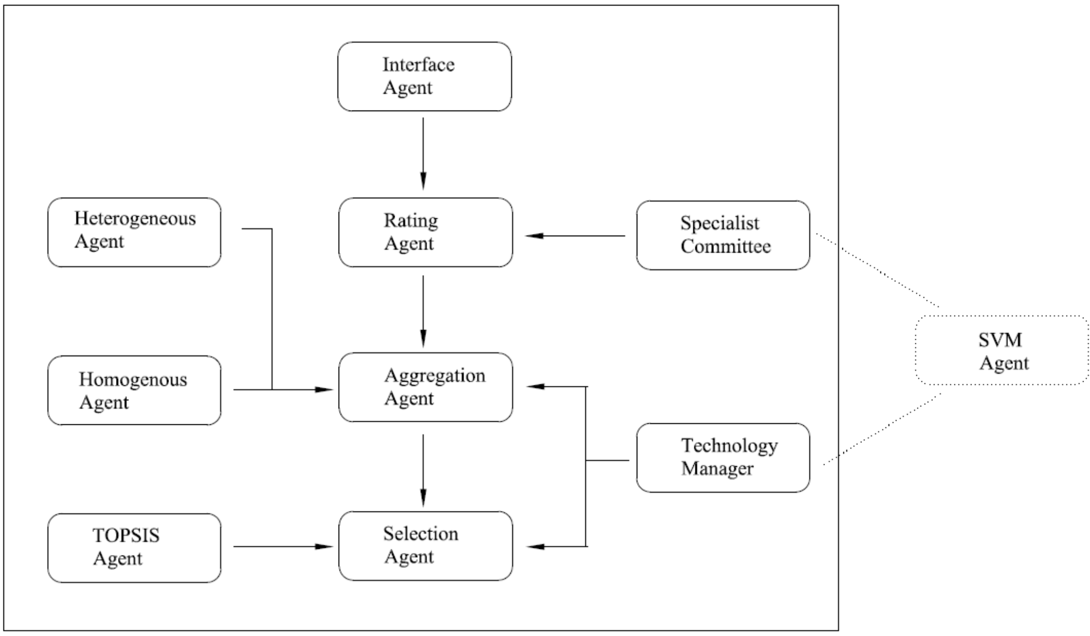


Figure 7.18 The learning approach agent group

The independent agents are the interface agent and output agent. The interface agent is the 'window' to the designers. The designers can set initial values though this agent. They can also control the process via this agent. The output agent takes charge of the output function.

7.6.3 Smart system environment

The smart environment in this system actually means the agent service platform. This environment looks like a virtual society as shown in Figure 7.19. It provides the information for every agent/agent group and also manages the output of the agent/agent group, which helps the agent/agent group to focus on its own inner actions. So the environment should have following functions:

The first one is the thread mechanism. The multi-agent has no priority for all agents and there is also no general manager agent to administer other agents. In order to make sure that the agent clearly knows what to do in the next step, an environment should be responsible for the management of design tasks in this system. The environment will provide an ID number to every design task and this unique ID number will lead the agent to work. There are many service nodes on every thread. The environment will force the thread to finish the service node one by one. On every service node, the thread will check the related agent or agent group. The agent/agent group will send the statement signal to the environment. The signal has three options: busy (Caprace, Losseau et al.), idle (green) and waiting (yellow). When the environment reads “busy” (Caprace, Losseau et al.), it will transfer to other agents and when this agent changes to red or yellow, thread will come back to try again after a specified time. When the environment reads idle (green), the environment will collect the information and give it to agent. The agent will be activated. When the environment reads waiting (yellow), the environment will hold a short time and at the same time, collect the data for further application. So the actions of design work are processed by the environment.

The second is the information sharing mechanism. In this system, all the information is open to public. This means that before taking actions, the agent will collect the necessary information from environment and after the run, the agents will directly provide the results to the environment. These results can be seen by every agent in the environment. This raises two problems: how to effectively manage the information and when to delete the out of date data to release the space. For the first problem, a ‘blackboard’ mechanism is employed here. The results are stored according to the ID number and a service period number is added to the ID number.

The last one is the conflict resolution. In the multi-agent environment, when there are several design tasks being processed at the same time, how to coordinate the agents to finish the work becomes very important. This system accepts prior method. The case-based conflict resolution is still preferred but not the first choice as the random selection is the first choice.

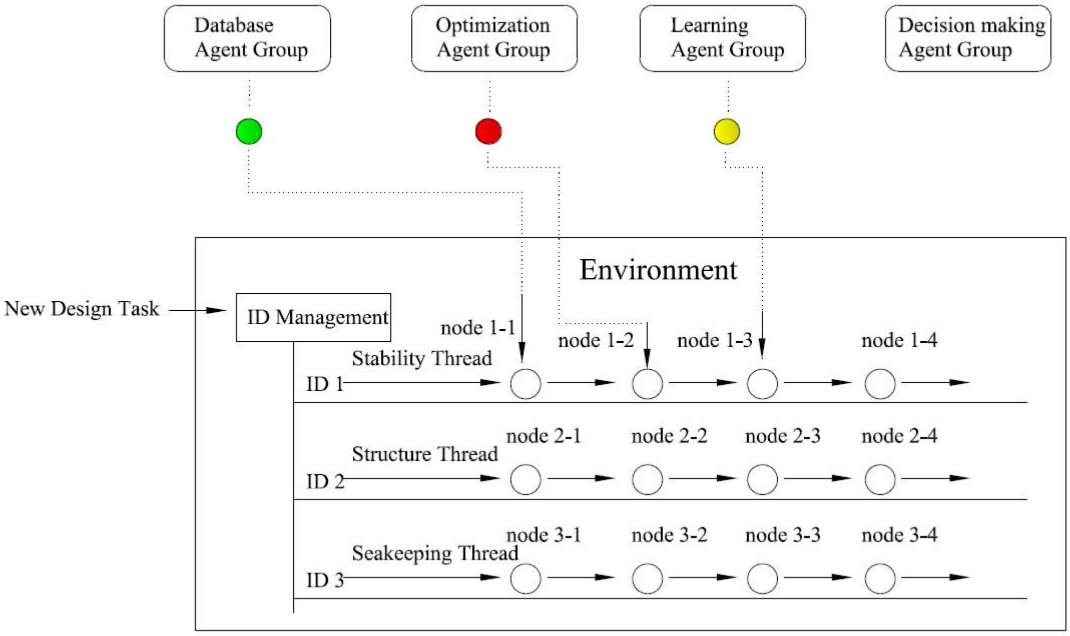


Figure 7.19 The smart environment of system

7.6.4 Software environment

The Java language is employed to build this design decision support system. The agent platform is constructed via the idea of JADEX, which is adopted from previous work (Turkmen 2005). But considering the future development, the software of JADEX is no longer used. The new platforms still accept XML to define the agent for user. The agent framework is employed to all agents. In other word, the agent is modular.

7.6 Discussion

In this chapter, the decision making method with learning function is constructed. The multi-agent system is employed and the SVM method is confirmed to realize the learning function. The designers can obtain an appropriate solution via this decision making method even if there is no specialists and technology manager. The virtual specialists and manager are created to give an evaluation on the optimisation results.

For the decision making method, the multi-agent framework makes the method robust and modular. This is also easy to accept new technology to update this method, for example, when a new ranking method is developed, the method can just change one agent to update without changing others. The agent based system also transfers original method to an automatic method and makes it more convenient to use.

The virtual specialist and manager use prior designs to make a prediction. This method can give a good prediction on a small sample. In the system, several kernel functions are provided to help the designers choose a good classification plane to perform a better prediction. In this chapter, the learning based ship design support system is concluded and the realization method and environment are explained. The utilization of the system is a hybrid process of the human and machine. The design decision support system can provide full and effective information to the designers. This design decision support system can successfully help the designers to solve the new design problem based on the prior experience.

Chapter 8

Case study

8.1 Introduction

This chapter applies the proposed system to practical ship design to evaluate the whole system. Two typical ship design problems are selected as case studies. The methods developed in this research are deployed to process design work and compared with other popular methods. The results are also compared against the original designs. The analysis and conclusion on application are given at the end of every case study.

Ship stability based subdivision optimisation is a traditional and classical problem in ship design. In all ship types, Ropax ships have particularly strict requirement for safety at sea. The case study 1 will apply the proposed system to hull subdivision design of Ropax ship. Ship structural optimisation is another important part of optimisation application in ship design. As other optimisation problems in this field, the time cost has a great influence on the structural optimisation. For a merchant ship in current computing environment, the ship structural optimisation usually continues for several weeks or months and this has become one of the main reasons that optimisation application has been limited. Case study 2 will deploy the proposed system for mid-ship structure optimisation of a bulk carrier in order to evaluate the performance of this system.

8.2 Case Study 1--- Safety design of Ropax ship

Ropax vessels have gone through significant design changes due to the well publicised recent accidents with the loss of many lives, while the demand for passenger and cargo capacity in European waters has increased. Following the tragic accidents of the Herald of Free Enterprise in 1987 and the Estonia in 1994, a significant surge of research related to the capsizing of Roll on –Roll off type ships was initiated. The 'Water on Deck' standards, which are known as Stockholm Agreement determines the limiting wave height（Hs）in which a ROPAX vessel survives in a damaged condition. In response to these regulations, the shipping industry has been in search of new modern designs to match these high safety standards, while maximizing the cargo capacity of vehicles in a cost effective approach.

The changes in design focus on the damage stability and survivability, cargo and passenger capacity. Therefore, the internal hull subdivision layout is an important problem especially for damage stability, survivability, internal cargo capacity and the general arrangement of the vessel.

In literature, different solution methods have been proposed. Sen and Gerick(1992) (Sen and Gerigk 1992) suggested using a knowledge-based expert system for subdivision design using the probabilistic regulations for passenger ships. Zaraphonitis et al. (Zaraphonitis, Boulougouris et al. 2003) proposed an approach for the optimisation of Ro-Ro ships in which centre-casing, side-casing, bulkhead deck height and locations of transverse bulkheads are treated as optimisation variables.

Ölçer et al. (Ölçer, Tuzcu et al. 2003) studied the subdivision arrangement problem of a ROPAX vessel and evaluated conflicting designs in a totally crisp environment where all the parameters are deterministic. They also examined the same case study in a fuzzy multiple attributive group decision-making environment, where multiple experts are involved and available assessments are imprecise and deterministic (Ölçer, Tuzcu et al. 2006). Turan et al. (2004) (Turan, Turkmen et al. 2004) approached the subdivision problem by using the case-based reasoning approach. Turkmen et al. (2005) (Turkmen 2005) proposed NSGAII with TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) to perform design optimisation for internal subdivision.

8.2.1 Problem modelling

The case study naturally focuses on improving the performance of the Ropax vessels in terms of not only maximising the ship related parameters mentioned above but also reducing the time to perform the multi-objective design iteration.

As this study is based on the previous work of Ölçer et al.(Ölçer, Tuzcu et al. 2003), (Ölçer, Tuzcu et al. 2006), the same problem is selected and used in this study. The optimisation problem is an internal hull subdivision optimisation for a ROPAX vessel whose main particulars are given in Table 8.1. Therefore the main aim of this case study is to maximise the survivability and damage stability standards as well as to improve the cargo capacity. These three parameters form the objectives of the study as presented in Table 8.2.

The recently developed 'Static Equivalent Method (SEM)' is used to calculate the limiting significant wave height (H_S) value for the worst SOLAS'90 damage, determined from damage stability calculations.

Stockholm Water on deck standards, which are applied specifically for Ropax vessels, determine the maximum significant wave height at which the Ropax vessel can survive and therefore is allowed to operate in wave heights, which are equal to or less than the maximum wave height. The Stockholm agreement allows model tests for demonstration of compliance to determine the maximum wave height as an equivalent approach. The SEM is an empirical capsize model for Ro-Ro ships that can predict with reasonable accuracy the limiting sea-state for specific damage conditions. The SEM for Ro–Ro ships postulates that the ship capsizes quasi-statically, as a result of an accumulation of a critical mass of water on the vehicle deck, the height of which above the mean sea surface uniquely characterises the ability of the ship to survive in a given critical sea state. This method was developed following observations of the behaviour of damage ship models in waves and it was validated using several model experiments and a large number of numerical simulations. Therefore in this particular case study SEM is used as an equivalent approach to the Stockholm Agreement.

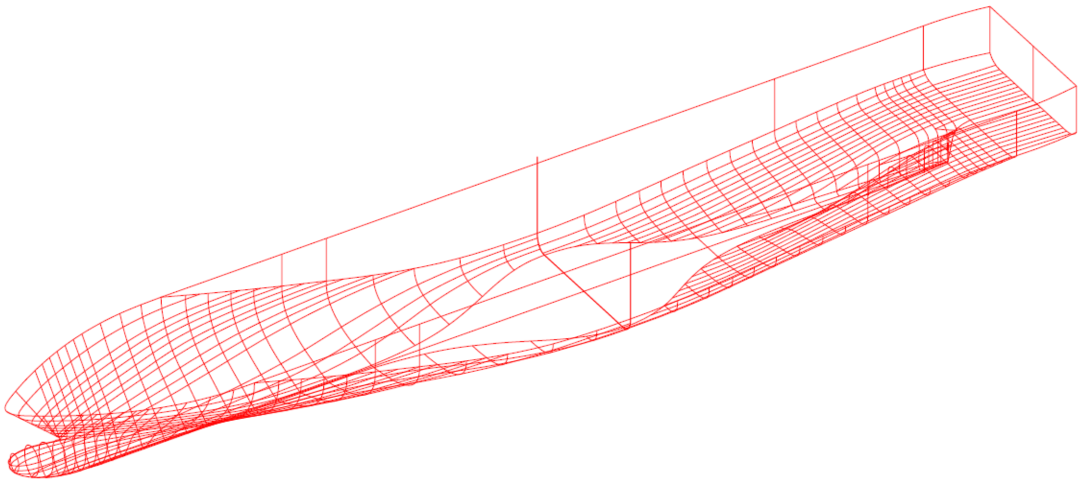


Figure 8. 1 Ship model built in NAPA for optimisation.

The whole ship has been modelled in NAPA software as shown in Figure 8.1 and the original ship hull subdivision is shown in Figure 8.2.

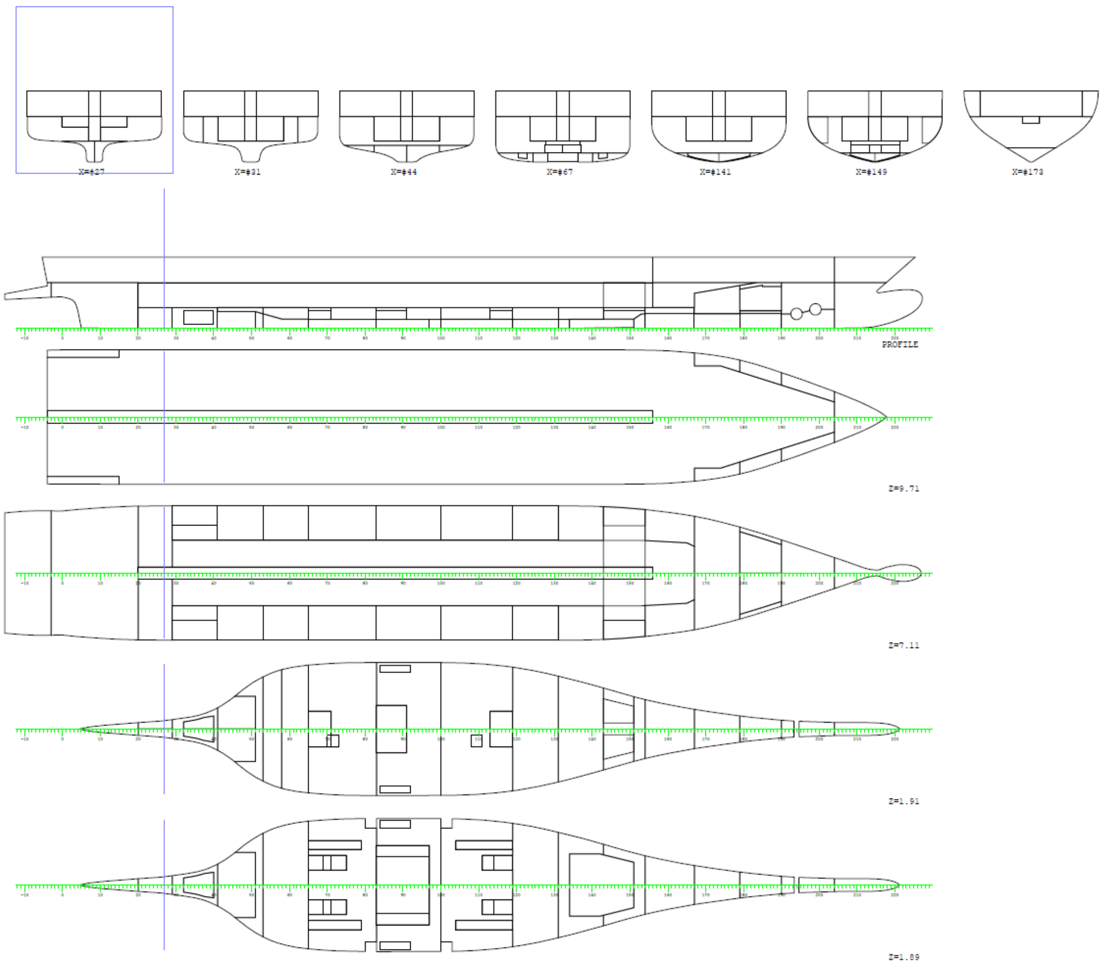


Figure 8. 2 The original ship hull subdivision model

Table 8. 1 Main dimensions of the vessel in case study.

Length Overall(L_{oa})	194.4 m
Length between perpendiculars (L_{bp})	172.2 m
Breadth moulded (B)	28.4 m
Depth to car deck	9.7 m
Lower Hold Height	2.6 m
Depth to upper deck	15.0 m
Draught design	6.6 m
Displacement	20200 ton
Max number of persons on board	2660
Number of car lanes	8

There are three objectives and 16 design variables in this study (Table 8.2). Three of the design variables are: the depth of the ship to the car deck, 'Car Deck Height', 'Lower Hold Height' and the width of the side casing at the car deck, 'Side Casing Width'. As presented, the original design has 9.7 m depth to car deck and 2.6 m lower deck height and no side casings at the car deck (Figure 8.6 a). The remaining 13 design parameters are the locations of transverse bulkheads given in the format of frame numbers (starting from the stern of the ship). Table 8.2 also presents the lower and upper boundaries of the parameters with assigned increments to be used in the optimisation study.

Table 8. 2 Optimisation variables with their types, bounds, and objectives.

No	Variables	Original design	Type		Bounds		Increment
			Discrete	Cont.	Lower	Upper	
1	Car deck height	9.7 m	√		9.6m	9.9m	0.025m
2	Lower-hold height	2.6m	√		2.6m	5.2m	2.6
3	Side casing width	No	√		1m	2m	0.5m
4	Transverse Bulkhead 02	27	√		25	29	1
5	Transverse Bulkhead 03	39	√		37	41	1
6	Transverse Bulkhead 04	51	√		49	53	1
7	Transverse Bulkhead 05	63	√		61	65	1
8	Transverse Bulkhead 06	81	√		79	83	1
9	Transverse Bulkhead 07	99	√		97	101	1
10	Transverse Bulkhead 08	117	√		115	119	1
11	Transverse Bulkhead 09	129	√		127	131	1
12	Transverse Bulkhead 10	141	√		139	143	1
13	Transverse Bulkhead 11	153	√		151	155	1
14	Transverse Bulkhead 12	165	√		163	167	1
15	Transverse Bulkhead 13	177	√		175	179	1
16	Transverse Bulkhead 14	189	√		187	191	1
Boundaries for transverse bulkheads are given in frame numbers							

No	Objectives	Type	Description
1	H_S value	Maximization	for the worst two compartment damage case
2	KG limiting value	Maximization	for the worst two compartment damage case
3	Cargo capacity value	Maximization	expressed in car lanes

The damage stability has been calculated according to the constraints from SOLAS'90 regulations as shown in Table 8.3.

Table 8. 3 Constraints of SOLAS'90 requirements

No	Constraints	**Requirements**
1	Range	Range of positive stability up to 10 degrees
2	Min. GZ Area	Minimum area of GZ-curve more than 0.015mrad
3	Maximum GZ	Maximum righting lever more than 0.1m
4	Maximum GZ due to Wind Moment	Maximum righting lever after applied wind moment more than 0.04m
5	Maximum GZ due to Passenger Moment	Maximum righting lever after passenger crowding more than 0.04m
6	Maximum Heel	Maximum static heel less than 12degrees
7	Minimum GM	Minimum GM more than 0.05m
8	Margin Line	Margin line should not be immersed
9	Progressive Flooding	No progressive flooding should occur

JAVA language is used to code the optimisation system according to the multi-agent structure and the calculation is processed between optimisation system and NAPA software. The Visual Basic (VB) is used to form the interface. The optimisation system uses NAPA to calculate the Hs, KG limiting and Cargo Capacity, and after optimisation, modified design variables are transferred to NAPA (Figure 8.3).

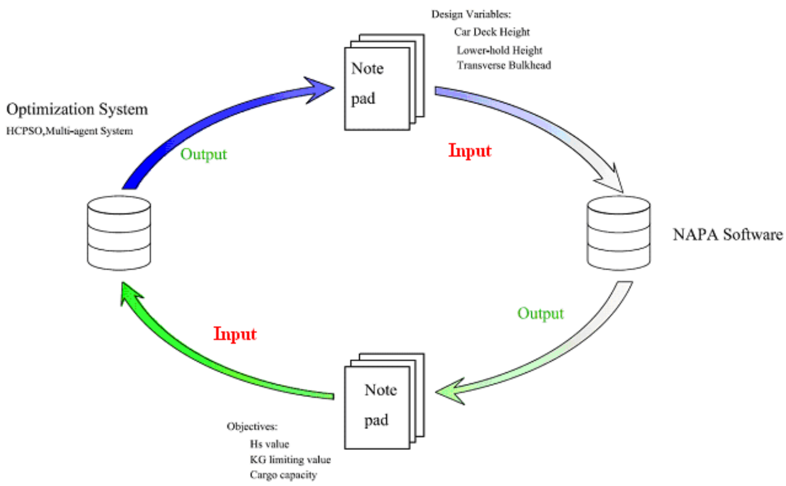


Figure 8. 3 Work flow between the Optimisation system and third party software NAPA

The vessel is modelled in NAPA and can be modified for each design experiment (or design layout) with respect to each optimisation parameter via NAPA macro language. For each design experiment, the relevant adjustments of Draught and Displacement are made during the optimisation process as shown in Figure 8.4.

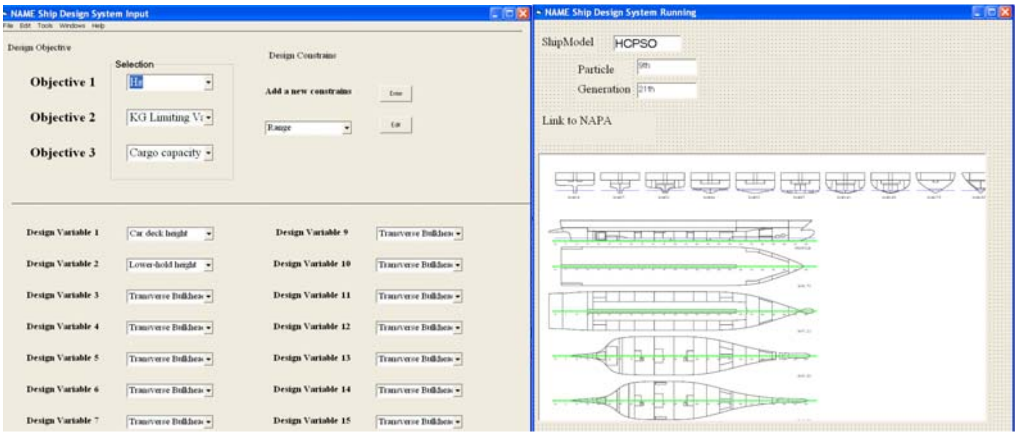


Figure 8. 4 Interface of ship design optimisation system.

8.2.2 Result and analysis of HCPSO

The calculations are performed using both NSGAII and HCPSO, and the results are compared in terms of numerical value of objectives and the computing time that each approach takes.

This optimisation has three objectives to maximize, thus three sub-swarms are set. The population size is 30 and generation is 100. Therefore every sub-swarm has 10 particles. c_1 and c_2 are set to 2.0. ω is gradually decreased from 1.0 to 0.4. Vmax is set to the bounds of decision variable ranges. χ is 0.72. The ε-disturbance has 3 steps. The NSGA II uses the parameters setting of prior research in the same environment (Turkmen 2005). The HCPSO and NSGAII parameters are listed in Table 8.4.

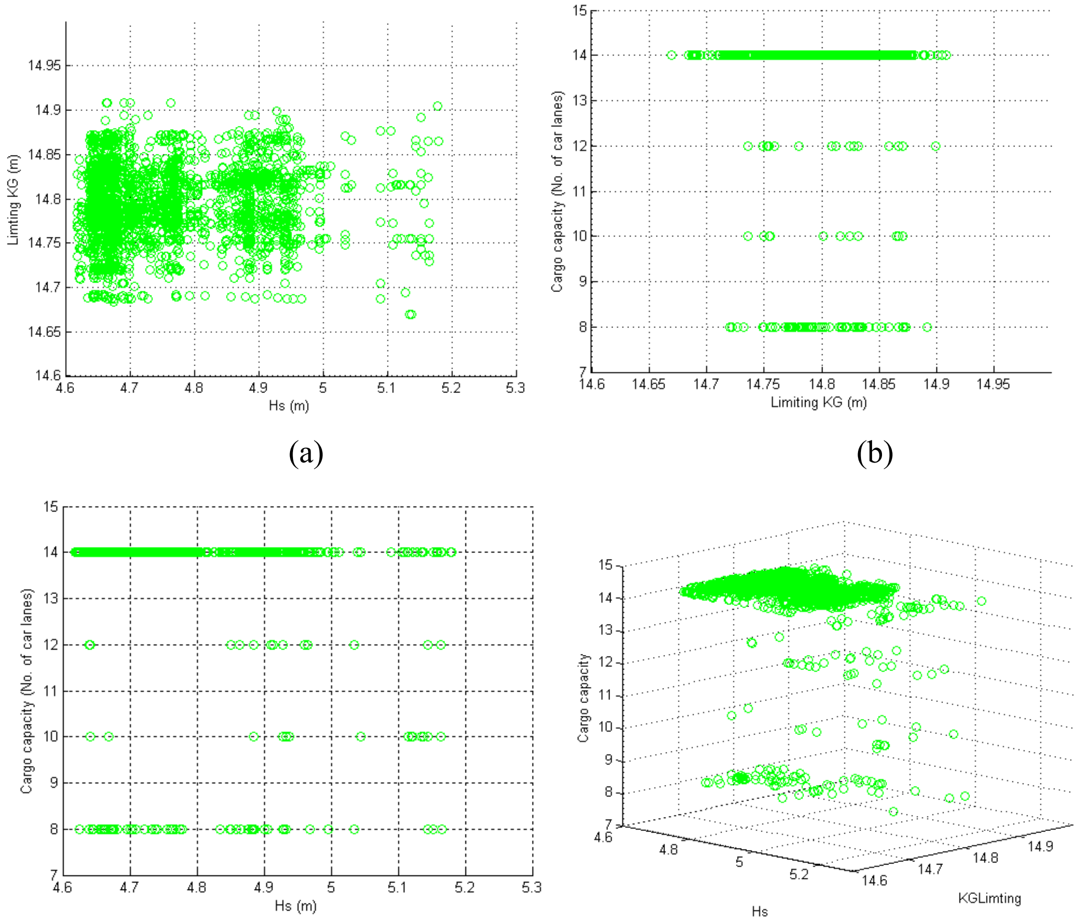
Table 8. 4 Parameters setting in case study for HCPSO and NSGAII

HCPSO Parameters Setting		**NSGAII Parameters Setting**	
Parameters Name	Parameters Value	Parameters Name	Parameters Value
Constriction Function	0.72	SBX (Simulated binary crossover)	10
Inertia weight	1.0 to 0.4	polynomial mutation	20
Cognitive parameter	2	crossover probabilities	0.9
Social parameter	2	mutation probabilities	0.1
Population	30	Population	30
Generation	100	Generation	100

The optimisation is performed using a PC (Dual Core 2.4GHz, 3 GB RAM) environment and takes 80h. At the end of this run, 2625 different designs are obtained in design space with 615 of them being unfeasible designs. Therefore 2010 (=2625-615) feasible designs are filtered in design space to obtain only the designs that belong to the Pareto front. The selected HCPSO optimisation solutions are listed in Table 8.5 together with the NSGAII optimisation. The comparison of original design and selected design in amidships is also given in Figure 8.6. In this solution (Table 8.5), the Hs in HCPSO is improved by more than 0.55 m compared to the original design, while the number of car lanes is increased by 6 extra lanes, which is equivalent to an increase of above 50%. The KG limiting value is also increased significantly (0.9 m) thus provides flexibility for future modifications on the basis of changing passenger demands as well as improvement in survivability. When comparing the two optimisation methods, the HCPSO design improves both the limiting KG and significant wave height (Hs) by 0.1m compared to the NSGAII design. Both methods provide the same solution for the cargo capacity by increasing the lower-hold height and car deck height, thus yielding more cargo capacity.

More importantly, for the real design case application, compared to NSGAII, HCPSO converges faster and reduces the computing, and hence the design time significantly. In ship design, most of the computing time is not consumed on the optimisation approach, but rather on the naval architectural calculations using third party software (NAPA etc.). It is important to note that it takes sometimes hours for one fitness calculation. In this study, the solution begins converging from 40th generations in HCPSO, while the NSGAII begins converging from 54th generations in NSGAII. This means the HCPSO takes 35% less time in looking for Pareto solutions compared to NSGAII. In a more complex environment, such as real ship application with many objectives, this provides an advantage in terms of completing the design faster and thus cheaper.

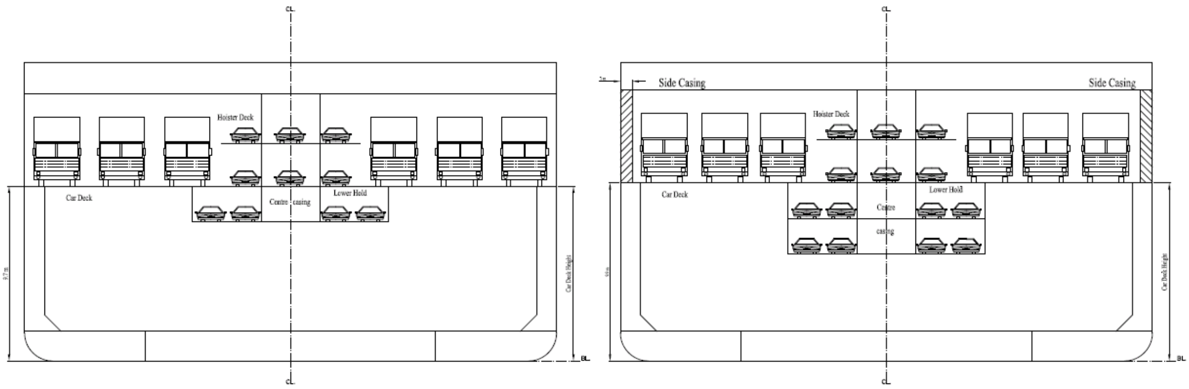
All feasible results are given in Figure 8.5. The relationships between different variables are shown for designers to make a direct observation and to select the appropriate solution according to their practical preference. All of the feasible designs are listed in 3D space (Figure 8.5 d).



(a) (b)

(c) (d)

Figure 8. 5 Optimisation results of HCPSO. (a) Limiting KG vs Hs; (b) Cargo Capacity vs Limiting KG; (c) Cargo Capacity vs Hs (d) optimisation feasible designs



(a) Original design (b) selected design

Figure 8. 6 Comparison between original design and selected design HCPSO.

Table 8. 5 Comparison of the original design and selected design which are optimal solutions for HCPSO and NSGAII

No	Optimisation Variables	Original Design	NSGAII Design	HCPSO design
1	Car deck height	9.7 m	9.9m	9.9m
2	Lower-hold height (from car deck)	2.6m	5.2m	5.2m
3	Side Casing width	No side-casing	1m	1m
	Watertight transverse bulkheads (In frame numbers)			
4	Transverse Bulkhead 02	27	27	27
5	Transverse Bulkhead 03	39	39	39
6	Transverse Bulkhead 04	51	52	53
7	Transverse Bulkhead 05	63	65	63
8	Transverse Bulkhead 06	81	83	83
9	Transverse Bulkhead 07	99	97	99
10	Transverse Bulkhead 08	117	115	118
11	Transverse Bulkhead 09	129	128	130
12	Transverse Bulkhead 10	141	141	143
13	Transverse Bulkhead 11	153	153	155
14	Transverse Bulkhead 12	165	164	167
15	Transverse Bulkhead 13	177	175	179
16	Transverse Bulkhead 14	189	189	189
	Optimisation Objectives			
1	H_S value (m)	4.641	5.082	5.179
2	KG limiting value (m)	14.012	14.813	14.9085
3	Cargo capacity value (lines)	8	14	14

8.2.3 Improving learning function of stability optimisation

In this part the same case of section **8.2.1** is selected to evaluate the Q learning in ship design. It is noteworthy that in the real case, the dynamic update is accepted which is obviously different with the numerical examples. This means, the system will use the calculation value to explore the beginning values group every time. The system will correct the wrong prediction value and recalculate the R value. Keep the continual circulation.

The optimisation results are listed in Table 8.6. This optimisation has three objectives to maximize, so three sub-swarm are set. In HCPSO, the population size is set to 30 and generation is set to 100. So every sub-swarm has 10 population. c_1 and c_2 are set to 2.0. ω is gradually decreased from 1.0 to 0.4. Vmax is set to the bounds of decision variable ranges. χ was 0.72. ε-disturb approach has 3 steps.

The results of HCPSO design and learning based HCPSO design are listed in Table 8.6. Compared to the original design, the limiting significant wave height that ship survives (Hs)by using learning based HCPSO, is improved by more than 0.5 m, while the number of car lanes is increased by 6 extra lanes, which is equivalent to almost 50% increase (Table 8.6). The KG limiting value is also increased significantly which provides flexibility for future modifications on the basis of changing passenger demands as well as increased survivability. Compared to the design based on HCPSO, the design based on HCPSO with learning, improved the Hs by 0.13m but limiting KG remains the same. This means that the improvement on the solutions of the HCPSO with learning to the HCPSO is not significant. However, the generation of learning HCPSO is significantly faster than normal HCPSO, which means the searching ability and speed of learning HCPSO is better than normal HCPSO. Time cost will be analysed in the following section.

Table 8. 6 Comparison of the Original design and selected design which are optimal solutions for HCPSO and Learning based HCPSO design

No	Optimisation Variables	Original Design	HCPSO design	Learning based HCPSO design
1	Car deck height	9.7 m	9.9m	9.9m
2	Lower-hold height (from car deck)	2.6m	5.2m	5.2m
3	Side Casing width	No side-casing	1m	1m
	Watertight transverse bulkheads (In frame numbers)			
4	Transverse Bulkhead 02	27	27	27
5	Transverse Bulkhead 03	39	39	40
6	Transverse Bulkhead 04	51	53	52
7	Transverse Bulkhead 05	63	63	63
8	Transverse Bulkhead 06	81	83	83
9	Transverse Bulkhead 07	99	99	100
10	Transverse Bulkhead 08	117	118	116
11	Transverse Bulkhead 09	129	130	131
12	Transverse Bulkhead 10	141	143	142
13	Transverse Bulkhead 11	153	155	155
14	Transverse Bulkhead 12	165	167	166
15	Transverse Bulkhead 13	177	179	177
16	Transverse Bulkhead 14	189	189	190
	Optimisation Objectives			
1	H_S value (m)	4.641	5.179	5.1921
2	KG limiting value (m)	14.012	14.9085	14.9126
3	Cargo capacity value (lines)	8	14	14

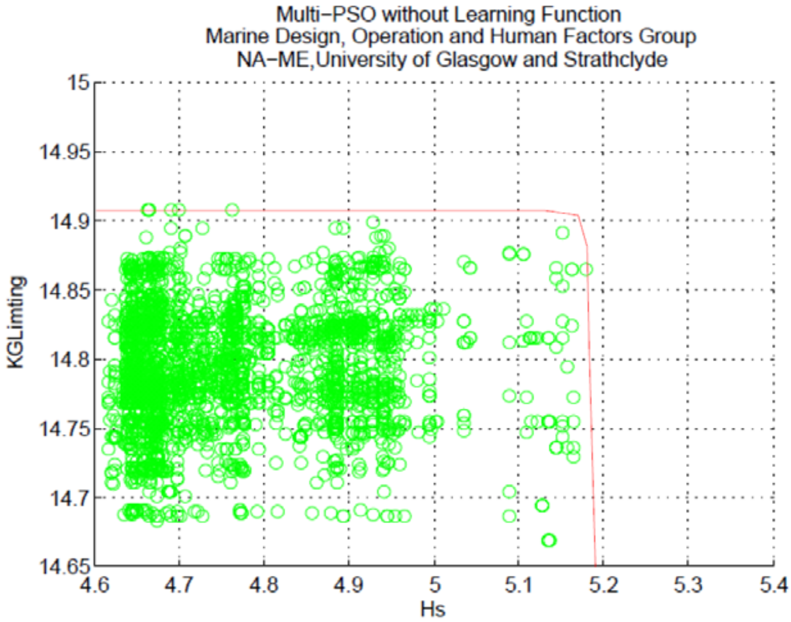


Figure 8. 7 KG limiting vs Hs HCPSO without learning function

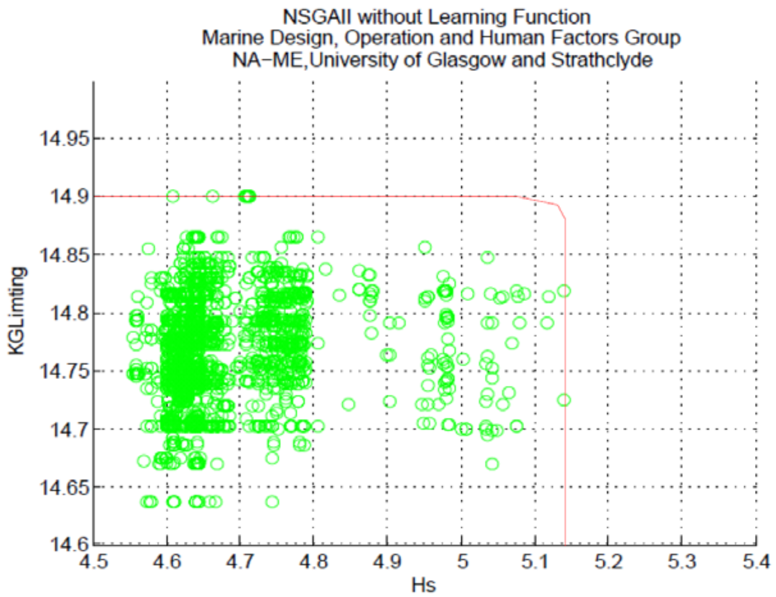


Figure 8. 8 KG limiting vs Hs NSGAII without learning function

In Figure 8.7 and 8.8, the red line is Pareto solution line. Comparing Figure 8.7 and 8.8, it can be seen that the HCPSO performs better than NSGAII from both the distribution of solutions and the numbers of solutions, which are close to Pareto solutions.

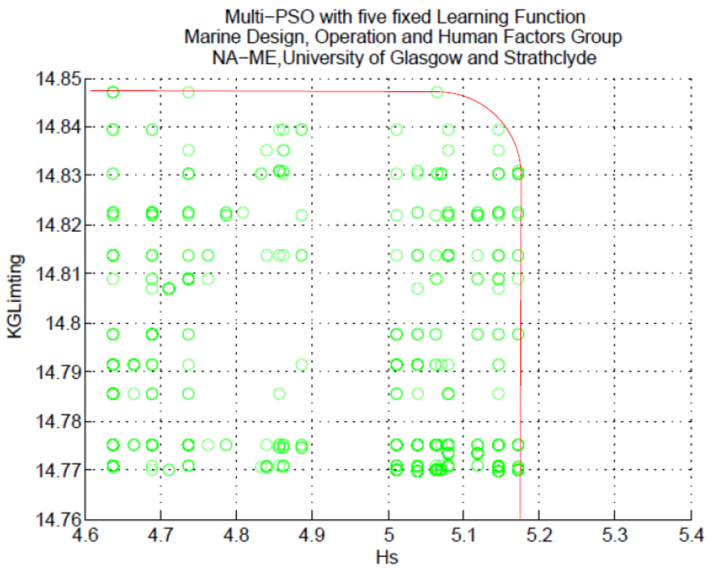


Figure 8. 9 KG limiting vs Hs HCPSO with fixed learning function

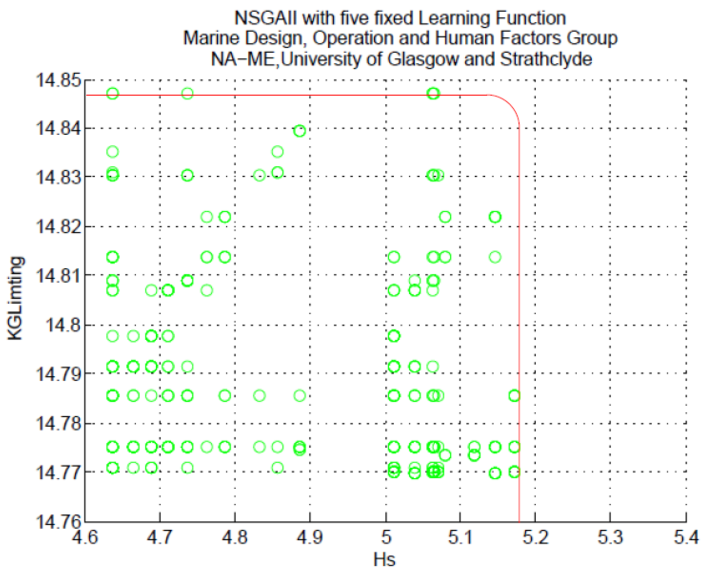


Figure 8. 10 KG limiting vs Hs NSGAII with fixed learning function

Comparing Figure 8.9 and 8.10, it can be seen that the HCPSO with fixed learning function performs better than NSGAII with fixed learning function on number of Pareto solutions. The red line is Pareto solution line.

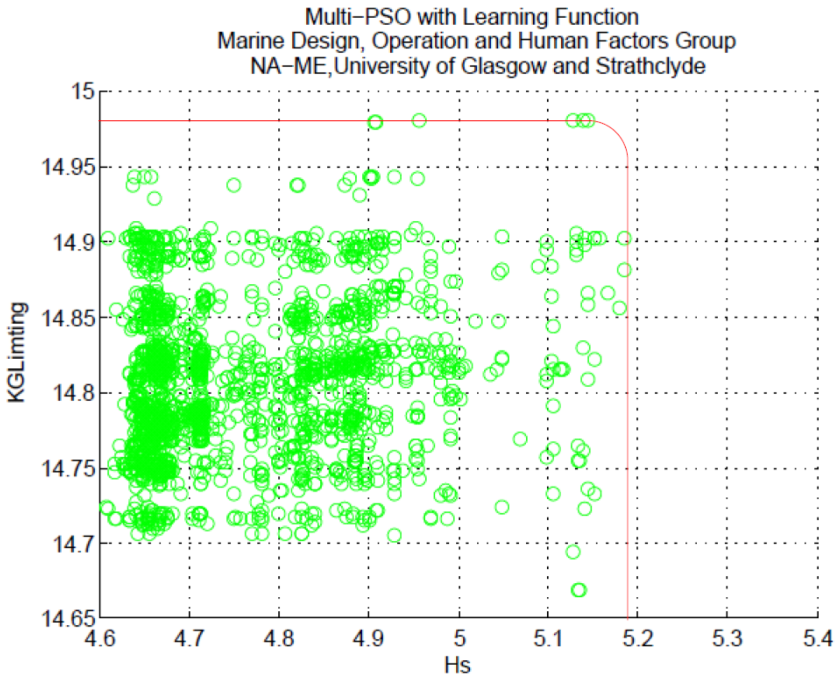


Figure 8. 11 KG limiting vs Hs HCPSO with learning function

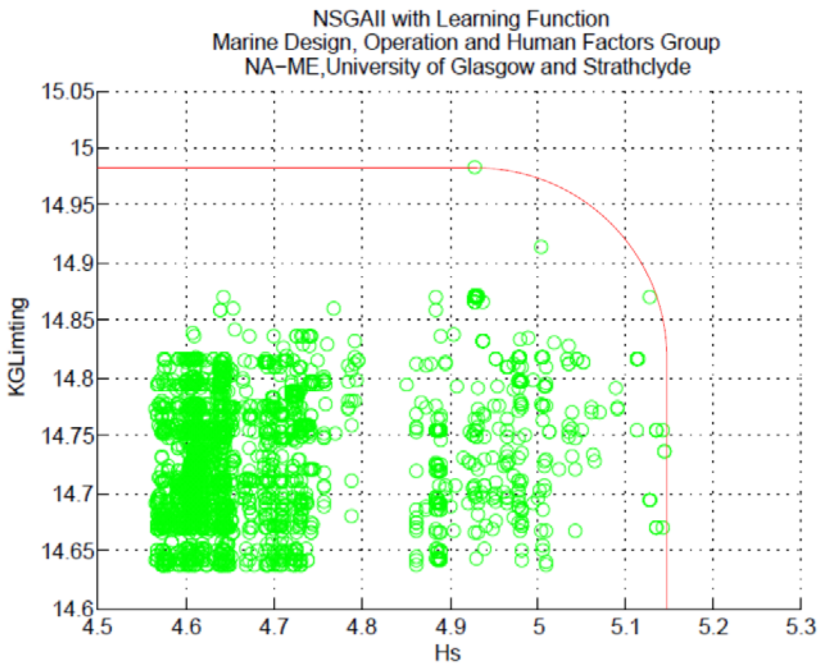


Figure 8. 12 KG limiting vs Hs NSGAII with learning function

Comparing Figure 8.11 and 8.12, it can be seen that the HCPSO with learning function performs better than NSGAII with learning function on number of Pareto solutions. The red lines provide Pareto solution lines for both algorithms. The numbers of Pareto solutions on the red line of HCPSO with learning are more than the Pareto solutions on the red line of NSGAII with learning function. At the same time, the solution distribution of HCPSO with learning function is more uniform. So the HCPSO with learning function gives more and better selections compared to NSGAII with learning function.

Time Cost

As far as the time cost is considered, most of the running time is voted by running NAPA simulation rather than optimisation. Therefore, reducing generations of convergence can reduce time. The summary of running time of both HCPSO and NSGAII is listed in Table 8.7. Figure 8.13 provides the comparison of run time of different HCPSO. And the Pareto solution of every HCPSO approaches. From Table 8.8, 8.9 and 8.10, it can be seen that the approach with learning function converge faster than other approaches. But for learning with five fixed one, which means just change five design variables every time, the solutions are not as good as random selecting one.

Table 8. 7 The summary of running time

Approach	Generation	Time (Hours)
HCPSO without learning	42-46	84-92
HCPSO learning with five fixed	15-24	30-49
HCPSO learning	25-35	60-80
NSGAII without learning	45-48	88-97
NSGAII learning with five fixed	20-28	40-56
NSGAII learning	32-46	64-90

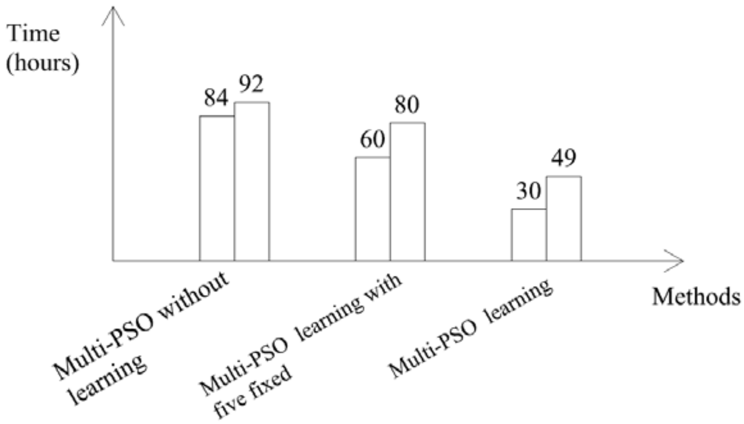


Figure 8. 13 Comparison of run time of different HCPSO

Table 8. 8 The Pareto solutions of HCPSO without learning

index	Generation	Hs	KG Lim.	Cap.	Time (h)
1	42	4.67835	14.9112	12	84
2	42	4.92334	14.9112	12	84
3	42	4.70002	14.9112	14	84
4	42	5.17285	14.8712	8	84
5	43	4.78847	14.9110	12	87
6	43	5.16283	14.8707	8	87
7	43	4.67835	14.9112	12	87
8	44	5.17285	14.8712	8	88
9	45	5.17285	14.8712	8	90
10	46	5.17285	14.8712	14	92
11	46	5.17285	14.8712	14	92

Table 8. 9 The Pareto solutions of HCPSO with 5 parameters fixed learning

index	Generation	Hs	KG Lim.	Cap.	Time(h)
1	15	4.63126	14.8471	14	30
2	15	5.15367	14.8393	8	30
3	17	4.73154	14.8471	8	33
4	18	5.17862	14.8321	12	36
5	18	4.73154	14.8471	8	36
6	19	5.17862	14.8091	8	38
7	19	5.17862	14.8238	8	38
8	19	5.17862	14.8227	8	38
9	19	5.17862	14.7712	8	38
10	23	5.17862	14.8142	12	46
11	24	5.17862	14.7753	12	49
12	24	5.17862	14.7712	8	49

Table 8. 10 The Pareto solutions of HCPSO with randomly selecting learning

index	Generation	Hs	KG Lim.	Cap.	Time
1	25	4.9210	14.9572	12	50
2	25	5.1387	14.9572	10	50
3	26	5.1465	14.9572	8	52
4	27	4.9223	14.9572	12	54
5	27	5.1428	14.9572	14	54
6	27	5.1921	14.8731	10	54
7	27	4.9670	14.9572	10	54
8	28	5.1921	14.9126	14	56
9	30	5.1503	14.9572	12	60

8.2.4 Decision making after optimisation

This section deploys the virtual committee, which is proposed in Chapter 7 to decide the final solution. For easier and better understanding, the Pareto solutions in the explanation part of this section are limited to two objectives: KG limiting and Hs.

First of all, the Pareto solutions of HCPSO without learning in Table 8.8 are considered. In the explanation part, there are only two objectives being considered. For simplified calculation, the solutions without objective 'cargo capacity' are listed in Table 8.11. The following calculations will be processed according to these solutions.

Table 8. 11 The simplified Pareto solutions of HCPSO without learning

Index	Generation	Hs	KG Lim.	Time (h)
1	42	4.67835	14.9112	84
2	42	4.92334	14.9112	84
3	42	4.70002	14.9112	84
4	42	5.17285	14.8712	84
5	43	4.78847	14.9110	87
6	43	5.16283	14.8707	87

The results of Table 8.8 are shown in Figure 8.14 using green colour while the original training data is represented by blue colour. The virtual expert E1, which has been created in chapter 7, is employed to make an evaluation on new solutions. From Figure 8.14, the results in different areas, which are divided by lines will be given as different expert opinions. The distinguishing method has been explained in section **7.5**. The proposed system will transfer these opinions to fuzzy numbers automatically. The linguistics term and their corresponding fuzzy numbers and membership function are shown in Table 8.12, which is conversed from Figure 7.4. The attribute's properties and weights of attributes and the values by experts are given in Table 7.3. The evaluations of virtual expert E1 are listed in Table 8.13.

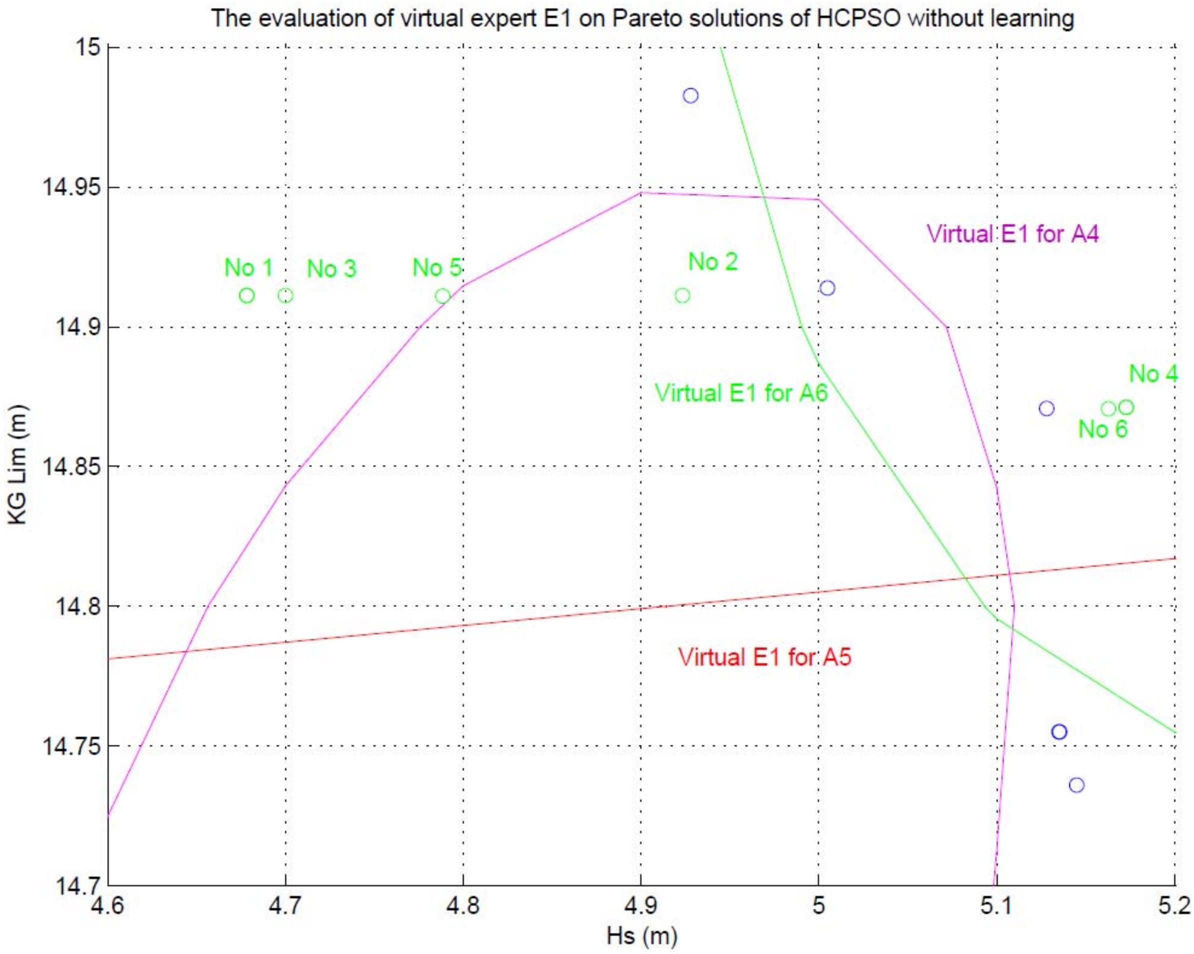


Figure 8. 14 Pareto solutions of HCPSO without learning on virtual expert E1

The results of evolutions of Pareto solutions of HCPSO without learning on virtual expert E1 are listed in Table 8.12.

Table 8. 12 Linguistics term and their corresponding fuzzy numbers and membership functions

Linguistic terms	Abbreviation	Corresponding Fuzzy Number
Bad	B	Trap(0.1,0.3,0.3,0.5)
Good	G	Trap(0.5,0.7,0.7,0.9)

Table 8. 13 Results of evolutions of Pareto solutions of HCPSO without learning on virtual expert E1

	Solution No	1	2	3	4	5	6
A1	numerical value	4.67835	4.92334	4.70002	5.17285	4.78847	5.16283
A2	numerical value	14.9112	14.9112	14.9112	14.8712	14.911	14.8707
A3	numerical value	N/A	N/A	N/A	N/A	N/A	N/A
A4	Expert opinions	Bad	Good	Bad	Bad	Bad	Bad
	Fuzzy numbers	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)
A5	Expert opinions	Good	Good	Good	Good	Good	Good
	Fuzzy numbers	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)
A6	Expert opinions	Bad	Bad	Bad	Good	Bad	Good
	Fuzzy numbers	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)

The evaluations of virtual experts E2 and E3 are shown in Figure 8.15 and Figure 8.16 together with the results listed in Table 8.14 and 8.15.

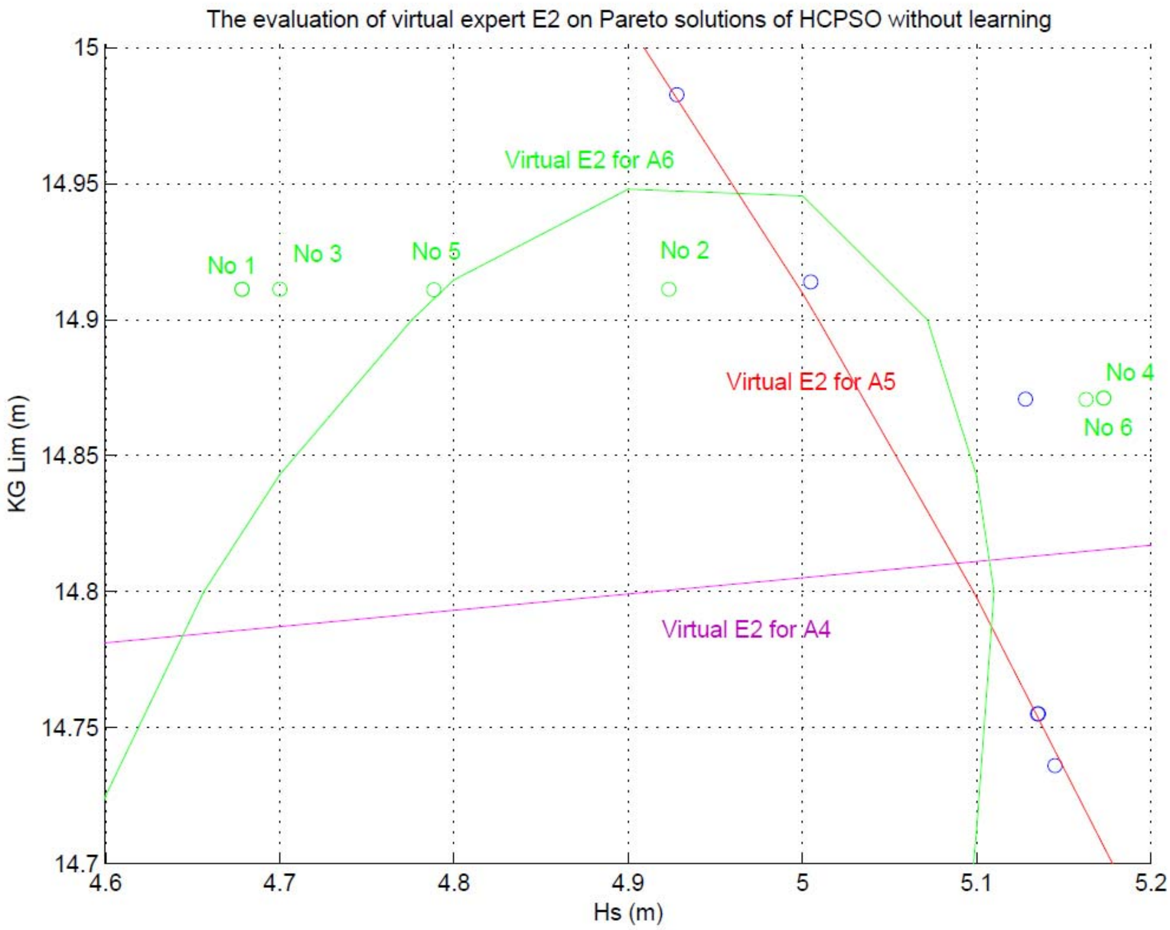


Figure 8. 15 Pareto solutions of HCPSO without learning on virtual expert E2

Table 8. 14 Results of evolutions of Pareto solutions of HCPSO without learning on virtual expert E2

	Solution No	1	2	3	4	5	6
A1	numerical value	4.67835	4.92334	4.70002	5.17285	4.78847	5.16283
A2	numerical value	14.9112	14.9112	14.9112	14.8712	14.911	14.8707
A3	numerical value	N/A	N/A	N/A	N/A	N/A	N/A
A4	Expert opinions	Good	Good	Good	Good	Good	Good
	Fuzzy numbers	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)
A5	Expert opinions	Bad	Bad	Bad	Good	Bad	Good
	Fuzzy numbers	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)
A6	Expert opinions	Bad	Bad	Bad	Good	Bad	Good
	Fuzzy numbers	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)

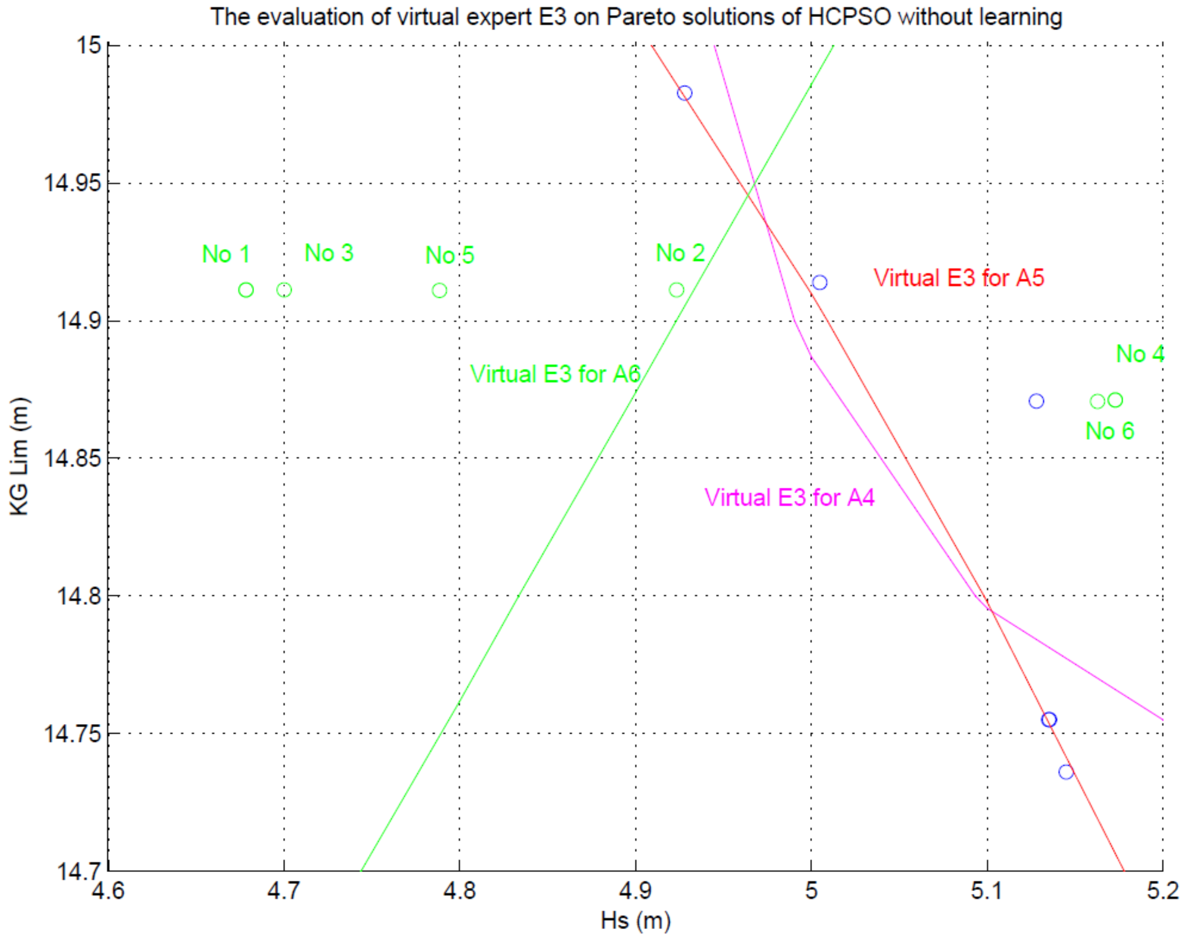


Figure 8. 16 Pareto solutions of HCPSO without learning on virtual expert E3

Table 8. 15 Results of evolutions of Pareto solutions of HCPSO without learning on virtual expert E3

Solution No		1	2	3	4	5	6
A1	numerical value	4.67835	4.92334	4.70002	5.17285	4.78847	5.16283
A2	numerical value	14.9112	14.9112	14.9112	14.8712	14.911	14.8707
A3	numerical value	N/A	N/A	N/A	N/A	N/A	N/A
A4	Expert opinions	Good	Good	Good	Bad	Good	Bad
	Fuzzy numbers	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)	(0.1,0.3,0.3,0.5)
A5	Expert opinions	Bad	Bad	Bad	Good	Bad	Good
	Fuzzy numbers	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)
A6	Expert opinions	Bad	Bad	Bad	Good	Bad	Good
	Fuzzy numbers	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)

Then the proposed system will automatically deploy the LMFMADM which is introduced in Chapter 7. Here, the main procedure in the system will be explained for further understanding. It should be kept in mind that in the proposed system, these calculations can be done by the system itself. The detailed algorithm is provided in appendix B.

Table 8.16, 8.17 and 8.18 calculate the aggregation of all three subjective attributes including the degree of agreement (or degree of similarity) (S), average degree of agreement (Aamodt), Relative degree of agreement (RA) and Consensus degree coefficient (CC).

Table 8. 16 Aggregation under the fourth attribute (A4)

Aggregation under the fourth attribute (A_4)

	Solution 1	Solution 2	Solution 3	Solution 4	Solution 5	Solution 6
E_1	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)
E_2	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)
E_3	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)	(0.1,0.3,0.3,0.5)
Degree of agreement (S)						
S_{12}	0.6	1	0.6	0.6	0.6	0.6
S_{13}	0.6	1	0.6	1	0.6	1
S_{23}	1	1	1	0.6	1	0.6
Average degree of agreement (AA)						
$AA(E_1)$	0.6	1	0.6	0.8	0.6	0.8
$AA(E_2)$	0.8	1	0.8	0.6	0.8	0.6
$AA(E_3)$	0.8	1	0.8	0.8	0.8	0.8
Relative degree of agreement (RA)						
$RA(E_1)$	0.273	0.333	0.273	0.364	0.273	0.364
$RA(E_2)$	0.364	0.333	0.364	0.273	0.364	0.273
$RA(E_3)$	0.364	0.333	0.364	0.364	0.364	0.364
Consensus degree coefficient (CC)						
$CC(E_1)$	0.324	0.360	0.324	0.378	0.324	0.378
$CC(E_2)$	0.378	0.360	0.378	0.324	0.378	0.324
$CC(E_3)$	0.298	0.280	0.298	0.298	0.298	0.298
$R^{HM}{}_{AG}$	(0.39,0.59,0.59,0.79)	(0.50,0.70,0.70,0.90)	(0.39,0.59,0.59,0.79)	(0.21,0.41,0.41,0.61)	(0.39,0.59,0.59,0.79)	(0.21,0.41,0.41,0.61)
$R^{HT}{}_{AG}$	(0.37,0.57,0.57,0.77)	(0.50,0.70,0.70,0.90)	(0.37,0.57,0.57,0.77)	(0.23,0.43,0.43,0.63)	(0.37,0.57,0.57,0.77)	(0.23,0.43,0.43,0.63)

Table 8. 17 Aggregation under the fifth attribute (A5)

Aggregation under the fifth attribute (A_5)

	Solution 1	Solution 2	Solution 3	Solution 4	Solution 5	Solution 6
E_1	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)	(0.5,0.7,0.7,0.9)
E_2	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)
E_3	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)
Degree of agreement (S)						
S_{12}	0.6	0.6	0.6	1	0.6	1
S_{13}	0.6	0.6	0.6	1	0.6	1
S_{23}	1	1	1	1	1	1
Average degree of agreement (AA)						
$AA(E_1)$	0.6	0.6	0.6	1	0.6	1
$AA(E_2)$	0.8	0.8	0.8	1	0.8	1
$AA(E_3)$	0.8	0.8	0.8	1	0.8	1
Relative degree of agreement (RA)						
$RA(E_1)$	0.273	0.273	0.273	0.333	0.273	0.333
$RA(E_2)$	0.364	0.364	0.364	0.333	0.364	0.333
$RA(E_3)$	0.364	0.364	0.364	0.333	0.364	0.333
Consensus degree coefficient (CC)						
$CC(E_1)$	0.224	0.224	0.224	0.260	0.224	0.260
$CC(E_2)$	0.358	0.358	0.358	0.340	0.358	0.340
$CC(E_3)$	0.418	0.418	0.418	0.400	0.418	0.400
$R^{HM}{}_{AG}$	(0.21,0.41,0.41,0.61)	(0.21,0.41,0.41,0.61)	(0.21,0.41,0.41,0.61)	(0.5,0.5,0.7,0.9)	(0.21,0.41,0.41,0.61)	(0.5,0.5,0.7,0.9)
$R^{HT}{}_{AG}$	(0.19,0.39,0.39,0.59)	(0.19,0.39,0.39,0.59)	(0.19,0.39,0.39,0.59)	(0.5,0.5,0.7,0.9)	(0.19,0.39,0.39,0.59)	(0.5,0.5,0.7,0.9)

Table 8. 18 Aggregation under the sixth attribute (A6)

Aggregation under the sixth attribute (A_6)

	Solution 1	Solution 2	Solution 3	Solution 4	Solution 5	Solution 6
E_1	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)
E_2	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)
E_3	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)	(0.1,0.3,0.3,0.5)	(0.5,0.7,0.7,0.9)
Degree of agreement (S)						
S_{12}	1	1	1	1	1	1
S_{13}	1	1	1	1	1	1
S_{23}	1	1	1	1	1	1
Average degree of agreement (AA)						
$AA(E_1)$	1	1	1	1	1	1
$AA(E_2)$	1	1	1	1	1	1
$AA(E_3)$	1	1	1	1	1	1
Relative degree of agreement (RA)						
$RA(E_1)$	0.333	0.333	0.333	0.333	0.333	0.333
$RA(E_2)$	0.333	0.333	0.333	0.333	0.333	0.333
$RA(E_3)$	0.333	0.333	0.333	0.333	0.333	0.333
Consensus degree coefficient (CC)						
$CC(E_1)$	0.280	0.280	0.280	0.280	0.280	0.280
$CC(E_2)$	0.320	0.320	0.320	0.320	0.320	0.320
$CC(E_3)$	0.400	0.400	0.400	0.400	0.400	0.400
$R^{HM}{}_{AG}$	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.5,0.5,0.7,0.9)	(0.1,0.3,0.3,0.5)	(0.5,0.5,0.7,0.9)
$R^{HT}{}_{AG}$	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.5,0.5,0.7,0.9)	(0.1,0.3,0.3,0.5)	(0.5,0.5,0.7,0.9)

After aggregation calculations, aggregation matrices for homo/heterogeneous group of experts can be constructed easily as shown in Table 8.19.

Table 8. 19 Aggregated matrices for homo/heterogeneous group of experts

Aggregated matrices for homo/heterogeneous group of experts

		Solution 1	Solution 2	Solution 3	Solution 4	Solution 5	Solution 6
A_1	Homo	4.67835	4.92334	4.70002	5.17285	4.78847	5.16283
	Hetero	4.67835	4.92334	4.70002	5.17285	4.78847	5.16283
A_2	Homo	14.9112	14.9112	14.9112	14.8712	14.911	14.8707
	Hetero	14.9112	14.9112	14.9112	14.8712	14.911	14.8707
A_3	Homo	12	12	14	8	12	8
	Hetero	12	12	14	8	12	8
A_4	Homo	(0.39,0.59,0.59,0.79)	(0.50,0.70,0.70,0.90)	(0.39,0.59,0.59,0.79)	(0.21,0.41,0.41,0.61)	(0.39,0.59,0.59,0.79)	(0.21,0.41,0.41,0.61)
	Hetero	(0.37,0.57,0.57,0.77)	(0.50,0.70,0.70,0.90)	(0.37,0.57,0.57,0.77)	(0.23,0.43,0.43,0.63)	(0.37,0.57,0.57,0.77)	(0.23,0.43,0.43,0.63)
A_5	Homo	(0.21,0.41,0.41,0.61)	(0.21,0.41,0.41,0.61)	(0.21,0.41,0.41,0.61)	(0.5,0.5,0.7,0.9)	(0.21,0.41,0.41,0.61)	(0.5,0.5,0.7,0.9)
	Hetero	(0.19,0.39,0.39,0.59)	(0.19,0.39,0.39,0.59)	(0.19,0.39,0.39,0.59)	(0.5,0.5,0.7,0.9)	(0.19,0.39,0.39,0.59)	(0.5,0.5,0.7,0.9)
A_6	Homo	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.5,0.5,0.7,0.9)	(0.1,0.3,0.3,0.5)	(0.5,0.5,0.7,0.9)
	Hetero	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.1,0.3,0.3,0.5)	(0.5,0.5,0.7,0.9)	(0.1,0.3,0.3,0.5)	(0.5,0.5,0.7,0.9)

The assessments have been transformed into standardized trapezoidal fuzzy numbers and then aggregated under each subjective attribute. In order to rank the alternatives, aggregated matrices' fuzzy elements should be defuzzified. Defuzzified values for the homo/heterogeneous group of experts are shown in Table 8.20 and 8.21.

Table 8. 20 Defuzzified values, (weighted) normalised ratings for Homogeneous group of experts

		Solution 1	Solution 2	Solution 3	Solution 4	Solution 5	Solution 6
	Defuzzified aggregated values	4.67835	4.92334	4.70002	5.17285	4.78847	5.16283
A_1	Normalised ratings	0.3891	0.4095	0.3909	0.4302	0.3983	0.4294
	Weighted normalised ratings	0.0856	0.0901	0.0860	0.0947	0.0876	0.0945
	Defuzzified aggregated values	14.9112	14.9112	14.9112	14.8712	14.911	14.8707
A_2	Normalised ratings	0.4086	0.4086	0.4086	0.4075	0.4086	0.4075
	Weighted normalised ratings	0.0695	0.0695	0.0695	0.0693	0.0695	0.0693
	Defuzzified aggregated values	12	12	14	8	12	8
A_3	Normalised ratings	0.4364	0.4364	0.5092	0.2910	0.4364	0.2910
	Weighted normalised ratings	0.0960	0.0960	0.1120	0.0640	0.0960	0.0640
	Defuzzified aggregated values	0.59	0.7	0.59	0.41	0.59	0.41
A_4	Normalised ratings	0.4314	0.5118	0.4314	0.2998	0.4314	0.2998
	Weighted normalised ratings	0.0475	0.0563	0.0475	0.0330	0.0475	0.0330
	Defuzzified aggregated values	0.41	0.41	0.41	0.65	0.41	0.65
A_5	Normalised ratings	0.3328	0.3328	0.3328	0.5277	0.3328	0.5277
	Weighted normalised ratings	0.0366	0.0366	0.0366	0.0580	0.0366	0.0580
	Defuzzified aggregated values	0.3	0.3	0.3	0.65	0.3	0.65
A_6	Normalised ratings	0.2733	0.2733	0.2733	0.5921	0.2733	0.5921
	Weighted normalised ratings	0.0465	0.0465	0.0465	0.1007	0.0465	0.1007

Table 8. 21 Defuzzified values, (weighted) normalised ratings for Heterogeneous group of experts

		Solution 1	Solution 2	Solution 3	Solution 4	Solution 5	Solution 6
A_1	Defuzzified aggregated values	4.67835	4.92334	4.70002	5.17285	4.78847	5.16283
	Normalised ratings	0.3891	0.4095	0.3909	0.4302	0.3983	0.4294
	Weighted normalised ratings	0.0856	0.0901	0.0860	0.0947	0.0876	0.0945
A_2	Defuzzified aggregated values	14.9112	14.9112	14.9112	14.8712	14.911	14.8707
	Normalised ratings	0.4086	0.4086	0.4086	0.4075	0.4086	0.4075
	Weighted normalised ratings	0.0695	0.0695	0.0695	0.0693	0.0695	0.0693
A_3	Defuzzified aggregated values	12	12	14	8	12	8
	Normalised ratings	0.4364	0.4364	0.5092	0.2910	0.4364	0.2910
	Weighted normalised ratings	0.0960	0.0960	0.1120	0.0640	0.0960	0.0640
A_4	Defuzzified aggregated values	0.57	0.7	0.57	0.43	0.57	0.43
	Normalised ratings	0.4208	0.5168	0.4208	0.3175	0.4208	0.3175
	Weighted normalised ratings	0.0463	0.0569	0.0463	0.0349	0.0463	0.0349
A_5	Defuzzified aggregated values	0.39	0.39	0.39	0.65	0.39	0.65
	Normalised ratings	0.3235	0.3235	0.3235	0.5392	0.3235	0.5392
	Weighted normalised ratings	0.0356	0.0356	0.0356	0.0593	0.0356	0.0593
A_6	Defuzzified aggregated values	0.3	0.3	0.3	0.65	0.3	0.65
	Normalised ratings	0.2733	0.2733	0.2733	0.5921	0.2733	0.5921
	Weighted normalised ratings	0.0465	0.0465	0.0465	0.1007	0.0465	0.1007

Determination of the positive-ideal solution can easily be made by taking the largest element for each benefit attribute and the smallest element for each cost attribute. The negative-ideal solution is just the opposite formation of the positive-ideal solution. Positive and negative ideal solutions are given in Table 8.22 for homo/heterogeneous group of experts.

Table 8. 22 Positive and negative ideal solutions for homo/heterogeneous group of experts

Positive-ideal solution			Negative-ideal solution		
	Homo	Hetero		Homo	Hetero
A1	0.086	0.086	A1	0.0947	0.0947
A2	0.0693	0.0693	A2	0.0695	0.0695
A3	0.064	0.064	A3	0.112	0.112
A4	0.033	0.0349	A4	0.0563	0.0569
A5	0.0366	0.0356	A5	0.058	0.0593
A6	0.0465	0.0365	A6	0.1007	0.1007

Table 8.23 and 8.24 show the values of separation measures and relative closeness to the positive-ideal solution for homo/heterogeneous group of experts. For homogeneous group of experts, the preference order is:
Solution 6 > Solution 4 > Solution 3 > Solution 5 > Solution 1 > Solution 2

Similarly, for heterogeneous group of experts, ranking is
Solution 3 > Solution 5 > Solution 1 > Solution 2 > Solution 4 = Solution 6

It should be kept in mind that if one may admit that the various experts are not equally important (or reliable), it is called heterogeneous (nonhomogeneous) group of experts and, otherwise, is named homogeneous group of experts.

Table 8. 23 Values of separation measures and relative closeness to the positive-ideal solution for homogeneous group of experts

	Solution 1	Solution 2	Solution 3	Solution 4	Solution 5	Solution 6
s_i^*	0.03513	0.039808	0.050147	0.058892	0.035165	0.058866
s_i^-	0.034407	0.038739	0.049643	0.059295	0.034442	0.059268
C_i^*	0.4948	0.4932	0.497475	0.501703	0.49481	0.501705
Rank	5	6	3	2	4	1

Table 8. 24 Values of separation measures and relative closeness to the positive-ideal solution for heterogeneous group of experts

	Solution 1	Solution 2	Solution 3	Solution 4	Solution 5	Solution 6
s_i^*	0.0618296	0.0605111	0.0602027	0.0507326	0.0615654	0.0507331
s_i^-	0.0624119	0.0610021	0.0608005	0.0509716	0.0621502	0.0509721
C_i^*	0.5023432	0.5020206	0.5024703	0.5011747	0.5023632	0.5011747
Rank	3	4	1	5	2	5

8.3 Case Study 2 --- Structure optimization

In this case study, the structural optimisation on the midship section of a bulk carrier was carried out. The objectives of this practical optimisation focus on the structural 17 weight controls and fatigue coefficients. The optimisation constraints are set according to common structural rules (CSR) of international association of classification societies while the stress and fatigue evaluation is also processed according to the methods in CSR.

Rigo (Rigo 2003) published a detailed state of the art paper in 2003 on structural optimisation research field. He introduced the concept and development of ship structure optimisation from the 1960s to 2003. In the same paper, Rigo introduced an optimisation software LBR-5 while Richir et al. (Richir, Caprace et al. 2007) used software LBR-5 (Rigo 2003) to solve three objectives optimisation problem. The production cost, weight and moment of inertia were selected as objectives and a two–stage local search heuristic approach (CONLIN) was accepted as an optimisation algorithm. Zanic et al. (Zanic, Andric et al. 2007) introduced a decision support methodology including optimisation for a multi-deck ship structure. Klanac (Klanac and Jelovica 2009) proposed vectorisation and constraint grouping approaches to enhance a fast ferry structure optimisation. Klanac (Klanac, Ehlers et al. 2009)introduced a two-stage optimisation approach to collision simulation. Eamon and Rais-Rohani (Eamon and Rais-Rohani 2009) presented a reliability-based optimisation method to a composite advanced submarine sail structure. Jang (Jang, Ko et al. 2009) employed a multi-objective genetic algorithm (MOGA) to solve a two objectives optimisation problem. Sekulski (Sekulski 2009) used a genetic algorithm to solve the problem of weight minimization of a high speed vehicle-passenger catamaran structure.

This study will use the hybrid algorithm of HCPSO and Q-learning to provide a new and fast optimisation method for structural designs.

8.3.1 The ship model used in this study

A bulk carrier is selected as the case study and the mid-ship structural will be optimisation according to new CSR rules. The main dimensions of this ship model are listed in Table 8.25. There are two objectives in this optimisation: weight control and fatigue. These objectives will be explained in section 8.3.3.

Table 8. 25 Main dimensions of proposed bulk carrier

Main Dimensions		
Length	180	m
Breadth	30	m
Depth to Up. Dk.	16	m
Design Draft	10.8	m
Scantling Draft	12.2	m
CB	0.85	
Speed	14	kn

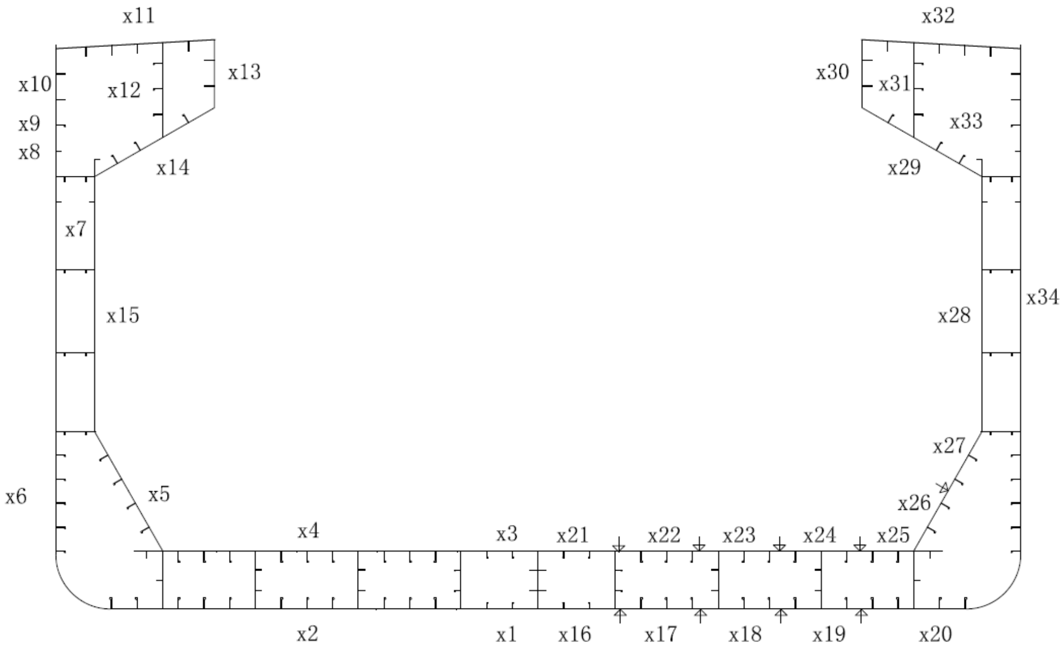


Figure 8. 17 The design variables of mid-ship structure

Figure 8.17 shows the design variables of mid-ship structure. There are 34 design variables while the design variables from x1 to x15 are the size of longitudinal stiffeners and from x16 to x34 are the shell thicknesses in mm. The Table 8.26 lists the minimum and maximum of design variables together with the changing increments. The upper bound of the longitudinal is 400mm and the lower bound is 100mm. The increment is 20 mm. The upper bound of the shell thickness is 10mm and the lower bound is 30mm. The increment is 1 mm.

Table 8. 26 Optimisation variables with types, bounds and increment

Design variables		Boundaries		Increment (mm)	Original Design
		Lower (mm)	Upper (mm)		
x1	longitudinal stiffener	150	350	10	220
x2	longitudinal stiffener	200	400	10	300
x3	longitudinal stiffener	150	350	10	240
x4	longitudinal stiffener	200	400	10	300
x5	longitudinal stiffener	200	400	10	340
x6	longitudinal stiffener	150	350	10	280
x7	longitudinal stiffener	100	200	10	150
x8	longitudinal stiffener	100	200	10	150
x9	longitudinal stiffener	200	400	10	320
x10	longitudinal stiffener	200	400	10	320
x11	longitudinal stiffener	200	400	10	340
x12	longitudinal stiffener	200	400	10	260
x13	longitudinal stiffener	200	400	10	320
x14	longitudinal stiffener	200	400	10	300
x15	longitudinal stiffener	100	200	10	150
x16	shell thickness	10	30	1	25
x17	shell thickness	10	30	1	25
x18	shell thickness	10	30	1	22
x19	shell thickness	10	30	1	22
x20	shell thickness	10	30	1	22
x21	shell thickness	10	30	1	25
x22	shell thickness	10	30	1	25
x23	shell thickness	10	30	1	22
x24	shell thickness	10	30	1	22
x25	shell thickness	10	30	1	22
x26	shell thickness	10	30	1	15
x27	shell thickness	10	30	1	15
x28	shell thickness	10	30	1	15
x29	shell thickness	10	30	1	15
x30	shell thickness	10	30	1	15
x31	shell thickness	10	30	1	15
x32	shell thickness	10	30	1	15
x33	shell thickness	10	30	1	15
x34	shell thickness	10	30	1	22

For this calculation, the professional CAE software ABAQUS is utilised to simulate and evaluate the designs while computer language JAVA is used as bridge to link the ABAQUS and optimisation algorithm. The CAE model is built in ABAQUS (as shown in Figure 8.18).

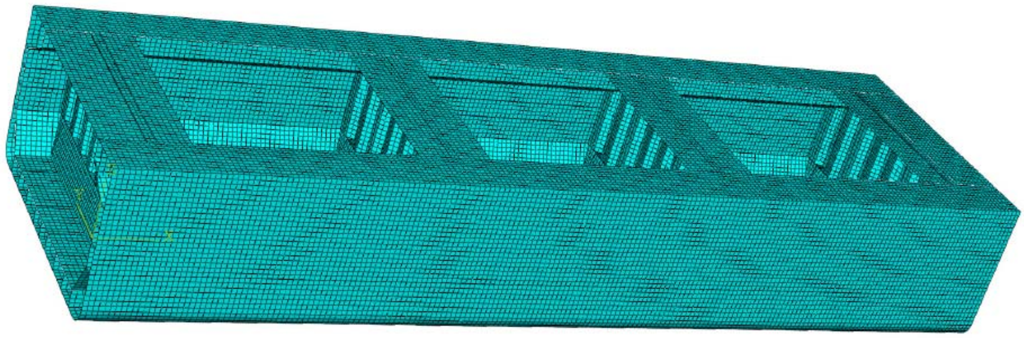


Figure 8. 18 The design variables of mid-ship structure

8.3.2 The simulation and calculation process

8.3.2.1 Objective 1 weight control

There are twenty-four situations, which need to be checked against the Common Structural Rules (CSR). The direct strength analysis will use a probability level of 10^{-8}, and the structure also will take into account of static loads but not dynamic loads. In this study, the standard density of steel is to be taken as 7.85 t/m^3 according to CSR. The Flowchart of FE analysis procedure follows the requirement of CSR as shown in Figure 8.19.

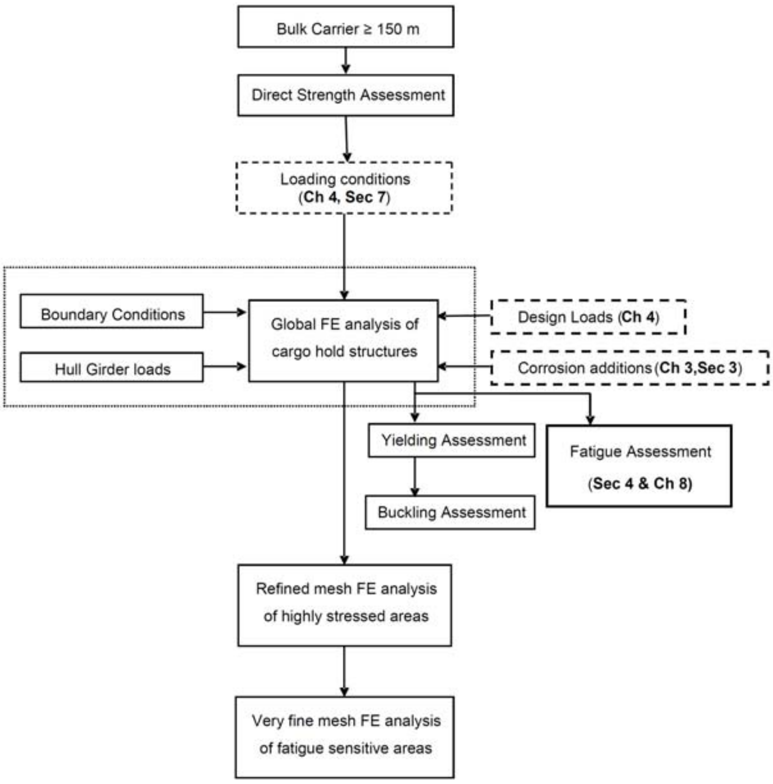


Figure 8. 19 FE analysis procedure (taken from CSR)

8.3.2.2 Objective 2 fatigue damage

The fatigue objective selects the joint part of bottom longitudinal and transverse bulkhead, which is also one of the key points of ship bulk carrier as the checking point. In this part, the new rules of common structural rules for bulk carriers of international association of classification society, which is effective from July 2009, is accepted. For simplified calculation, only one point as hot point is selected in this study. The cumulative fatigue damage D calculated for the combined equivalent stress should comply with the following criteria:

$$D = \sum_j D_j \le 1.0$$

Where

D_j : Elementary fatigue damage for each loading condition "j".

8.3.2.3 Constraints in structural optimisation

Boundary conditions

The boundary conditions are simply supported in both ends of the model according to Table 8.27 and Table 8.28.

Table 8. 27 Rigid-link of both ends (taken from CSR)

Nodes on longitudinal members at both ends of the model	Translational			Rotational		
	Dx	Dy	Dz	Rx	Ry	Rz
All longitudinal members	RL	RL	RL	-	-	-
RL means rigidly linked to the relevant degrees of freedom of the independent point						

Table 8. 28 Support condition of the independent point (taken from CSR)

Location of the independent point	Translational			Rotational		
	Dx	Dy	Dz	Rx	Ry	Rz
Independent point on aft end of model	-	Fix	Fix	Fix	-	-
Independent point on fore end of model	Fix	Fix	Fix	Fix	-	-

$$M_{\mathrm{SW\_S\_H1\ full\ load}} = 1.5\times10^{6} \quad \mathrm{kN\cdot m}$$

$$M_{\mathrm{WV\_H\_H1\ full\ load}} = 1.48\times10^{6} \quad \mathrm{kN\cdot m}$$

$$M_{\mathrm{WV\_S\_H1\ full\ load}} = 1.56\times10^{6} \quad \mathrm{kN\cdot m}$$

Other constraints are accepted from CSR.

8.3.3 Runs and results

The Parameters setting in case study for NSGAII and learning based NSGAII is set according to Table 8.29.

Table 8. 29 Parameters setting in case study for NSGAII

NSGAII Parameters Setting

Parameters Name	Parameters Value
SBX (Simulated binary crossover)	10
polynomial mutation	20
crossover probabilities	0.9
mutation probabilities	0.1
Population	30
Generation	100

The optimization algorithm is realized via JAVA computer language. The Figure 8.20 is the intermediate state of calculation in ABAQUS.

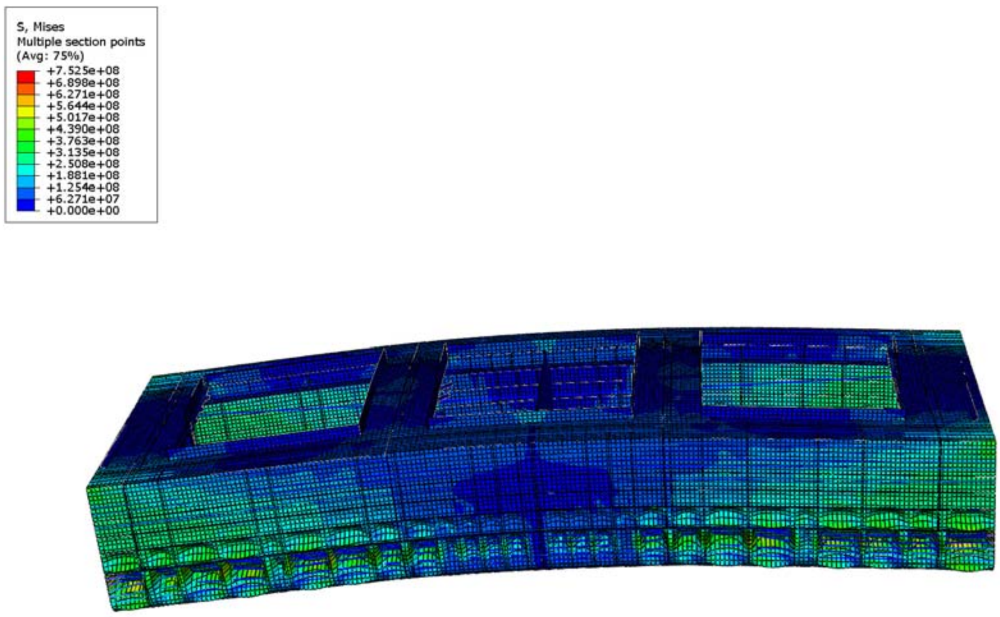


Figure 8. 20 The calculation in ABAQUS

Table 8. 30 Optimisation variables with their types, bounds and results

Design variables	Boundaries Lower (mm)	Upper (mm)	Increment (mm)	Original Design	NSGAII	Learning based NSGAII
x1	150	350	10	220	200	180
x2	200	400	10	300	280	280
x3	150	350	10	240	200	220
x4	200	400	10	300	350	320
x5	200	400	10	340	320	320
x6	150	350	10	280	240	220
x7	100	200	10	150	140	160
x8	100	200	10	150	160	160
x9	200	400	10	320	280	300
x10	200	400	10	320	260	260
x11	200	400	10	340	260	270
x12	200	400	10	260	240	220
x13	200	400	10	320	280	260
x14	200	400	10	300	280	280
x15	100	200	10	150	140	140
x16	10	30	1	25	22	20
x17	10	30	1	25	22	20
x18	10	30	1	22	18	19
x19	10	30	1	22	18	18
x20	10	30	1	22	18	19
x21	10	30	1	25	23	22
x22	10	30	1	25	22	20
x23	10	30	1	22	18	19
x24	10	30	1	22	18	19
x25	10	30	1	22	17	18
x26	10	30	1	15	16	15
x27	10	30	1	15	16	16
x28	10	30	1	15	15	15
x29	10	30	1	15	15	14
x30	10	30	1	15	15	16
x31	10	30	1	15	17	16
x32	10	30	1	15	15	15
x33	10	30	1	15	15	14
x34	10	30	1	22	20	20
Objective 1: Weight				45.78	32.17	31.89
Objective 2: Fatigue				0.812	0.723	0.704

The optimisation is performed using a PC (Dual Core 2.4GHz, 3 GB RAM) environment. At the end of this run, using proposed method, 3000 different designs are obtained in the design space with 812 of them being unfeasible designs. Therefore 2188 (=3000-812) feasible designs are filtered in design space of learning based NSGAII to obtain only the designs that belong to the Pareto front when for NSGAII, 2262 feasible designs are obtained. The selected solutions of the learning based NSGAII and NSGAII are listed in Table 8.30 together with the original design. From Table 8.30 it can be seen that the weight of the structure has been decreased significantly (more than 25%) while the fatigue coefficient decreased at the same time. More importantly, for the real design case application, compared to NSGAII, learning based NSGAII converges faster and reduces the computing, and hence the design time significantly. It is the same as for the stability calculation that the improvement between NSGAII and learning based NSGAII is not obvious but computation time changes significantly. In ship design, most of the computing time is not consumed on the optimisation approach, but rather on the naval architectural calculations using third party software. It is important to note that it takes sometimes hours for one fitness calculation. In this study, the solution begins converging from the 62th generations in learning based NSGAII, while the NSGAII begins converging from 84th generations in NSGAII. This means the proposed method takes 25% less time in looking for Pareto solutions compared to NSGAII. In a more complex environment, such as a real ship application with many objectives, this provides an advantage in terms of completing the design faster and thus cheaper.

8.4 Discussion

A subdivision design problem is selected as case study to evaluate this learning based support system. SOLAS is selected and three objectives are selected for optimisation. The study shows the system can draw experience from prior work and make a successful optimisation design. The final result has better performance comparing to original design.

With regards to the real ship design case, structural optimisation of a bulk carrier, with two conflicting objectives (weight and fatigue), is carried out. For the operation platform, a JAVA based optimisation system and ABAQUS has been integrated into optimisation framework.

The proposed algorithm provides improved design compared to the original design in every chosen objective with a significant margin and demonstrates the value of this method. In this design case, the proposed algorithm displays better performance both in speed and final results. The proposed algorithm is structured via a multi-agent system and every agent works remarkably well. It can be concluded that the proposed approach has shown great potential and can be applied to similar and even more complex optimisation problems in ship design, as well as to related areas within the maritime industry.

Chapter 9

Discussions and Conclusions

9.1 General

In this study, a systematic learning based decision support system is proposed and developed for ship decision practice. The research focuses on solving gaining experience /learning problem in the ship design work. The nature of ship design raises a requirement for sharing the experience. Current approaches for decision support system are based on simulation and performance evaluation of ship design but do not include gaining experience or learning in ship design. As a complex systematic engineering field, the design work depends largely on the experience of experts. This means that the quality of design is decided, to a great extent, by the specialists' personal skills. This makes it difficult for the development of a design support system and fast design. In this thesis, the design decision support system, which deploys the learning method to help the system to gain experience from previous cases is developed. The system can automatically apply these experiences to the design process and make predictions for the design problem.

In this thesis, many learning approaches including decision tree, CBR, Q-learning, SVM etc. are studied together. All of these approaches are composed of a systematic method for dealing with the learning problem in the ship design process. The research focuses on the experience database building, advanced algorithms on optimisation, learning based real time controlling in optimisation process and learning based decision making process. The research of the learning ability of ship design decision support system presented in this thesis is the first and it is expected to pave the way for future studies in this field. In view of the study undertaken the discussions are divided into several sections. In this chapter, the contributions and novelties are discussed while the encountered difficulties are presented. In the recommendation, the suggestions of design practice are discussed while a number of recommendations and further work considerations are suggested.

9.2 Key contributions and novelties

This study is the first to propose to deploy the machine learning method to assist the experience sharing in ship design. The different machine learning approaches are applied on different cases of the proposed system in ship design.

The proposed system changes the ship data storing model. It uses a hybrid database model which analyses the relationships and stores these relationships in order to replace the traditional data storing model. This new database model improves the efficiency of data utilization when it greatly reduces the task of designers.

A new optimisation method HCPSO is created. The HCPSO has excellent performance on both numerical functions and practical design with very easy parameter setting which leads to faster convergence. This research studies the performance of this new algorithm and presents the setting of the usual parameters of this algorithm.

In the proposed system, the Q-learning approach is to be applied for the first time onto the real time ship design optimisation. The run time of optimization is further reduced via intelligently predicted solution space. This study also presents the examples of combining the Q-learning with GAs and HCPSO.

This study successfully improves the traditional decision making method. It solves the automatic updating and the algorithm no longer depends on manual calculation and EXCEL tools, and the proposed system will calculate automatically. The Support Vector Machine is successfully imported into the decision making method in order to solve the issue of a lack/unavailability of expert.

The multi-agent idea is applied into the proposed system and a platform is developed for linking different third party software.

9.3 Discussions

In the light of the objectives outlined in Chapter 1, the research presented in this thesis focuses on the development of the learning based ship design decision system. The multi-agent framework is deployed to realize this system in a JAVA environment. The learning function of the system has been studied from theory to practice.

The SDLL as a new type of experience database is the basis of the whole system. The SDLL extends the function of storing the data to discovering all kinds of knowledge including implicit relationships from the previous cases. The attributes of the cases have been divided into two classes: numerical and linguistic. The corresponding learning approaches are given. The decision tree is selected to deal with the numerical attribute. The numerical attribute is created before it is determined and the tree is produced by the system together with the final results, when it is applied on real cases.

The CBR is employed to deal with the linguistic attribute, which would not be analyzed until the system needs to use it. In the study, a new method is created to measure the distance between different linguistic attributes. The linguistic attributes are divided into different parts for calculating the distance. XML is the data format of SDLL, which can automatically analyze the relationships of every attributes and mining these relationship as the design regulations.

The optimisation approaches in ship design are studied and a new HCPSO algorithm has been proposed and realized via a multi-agent framework. The HCPSO is one of the hybrid algorithms based on PSO and has a much quicker convergence rate than most other algorithms. For the optimisation in ship design, one of the most important application problems is the time cost. Since the evaluation of fitness function is processed via a third party software, this process will cost a lot of time. However, fast design is one of the main development directions for marine design, therefore it is necessary to have a new algorithm which can cost much less time to run and provides good interface for the learning method. The PSO algorithm has less parameters and better convergence than GA algorithm. The HCPSO inherits this performance of PSO and the less the number of parameter settings easier the controlling and embedding for the learning model.

The real time learning has been realized via Q-learning algorithm in this system. The discrete character of ship design is fully utilized into Q-learning algorithm. The system develops the real time learning as an independent module, so the optimisation algorithm can deploy it to improve the performance of different optimisation approaches. The combined applications of learning model and HCPSO, and Learning model and NSGAII have been introduced in Chapter 6. The comparison of original optimisation and optimisation approach with learning ability has been processed to show the advantages of the learning model. The results have shown that the optimisation with the learning model had a quicker convergence rate. Different look-up tables have been tested for comparing the efficiency.

The new integrated fuzzy multiple attribute decision-making (FMADM) approach with learning ability has been developed for the support system. The decision-making approach is rebuilt via a multi-agent framework. The new agent can deal with the problem independently and collaborate together to solve the .decision making problem. What is more, the rebuilding of approach makes this approach suitable for modularization. This makes the updating of the approach also very easy. The new integrated approach can deploy a different ranking model to improve the efficiency of ranking.

The specialist committee and technology manager are the key factors of the FMADM and decide the using efficiency of this method. This support system builds the virtual specialist committee and technology manager via learning approach to replace the human beings. The virtual specialist committee and technology manager can give the evaluation to the design solutions via the learning approach when the human specialists are absent. The SVM method was employed to deal with the small samples situation. The prior evaluations as examples are analyzed and used to find the relationships, and then when the new case comes, the system can give the prediction. The virtual specialist committee and technology manager can also reduce the time of decision making by calling human specialists, if they are available.

The JAVA language and multi-agent system framework are deployed to realize the whole support system. The smart environment was created via JAVA to simplify the system. The agent and agent group in the system are independent and there is no chief agent to control other agents. The agents make dialogue and information sharing via smart environment. The conflict solving follows the previous work. And the multi-task and parallel computing have been taken into account. It is proven that the system is suitable for concurrent engineering.

The subdivision design problem, one of the important design problems, was selected to evaluate the system. The system successfully found the solutions to the optimisation. The NAPA software as fitness function calculation tool is employed to make evaluations. A RORO ship was selected and new SOLAS 2009 was applied as design standards.

9.4 Recommendations

In the design and application of this learning based ship design decision support system, there are some discussions and considerations raised that will benefit the ship designers. These recommendations are also useful for the developers of support system for future research as well.

The partition of linguistic attributes should not be too small. On the other hand too large a span will reduce the practicality of the system on example cases. The distance of every attribute is systematically considered by the system. If there are weights for the attributes, the system can give correct evaluation to these attributes, but most of the time, there are no weights. So the designers should be more careful for linguistic attributes.

The initial value settings are important for the optimisation of PSO and GAs. In these study cases, all the applications are given in a random selection of design values. For the real ship design problems, the initial values were also selected via initial design. This means that the system selected the initial design and these randomly selected other designs via different design variables. In the engineering application area, there is another way to give initial values according to fitness function, and the different initial values would cause a big difference for optimisation results especially in engineering applications. One of the reasons is the engineering application sometimes can not get final Pareto global optimal solutions. Therefore the initial values play an important role on the outcome of the final optimisation solutions. The ship designers should try different initial values and select the best solutions.

The other problem is the running time and parameter settings. The GAs and PSO are all heuristic methods, which mean the performance depends on the population and generation. Different population and generation will cause different results. The parameter setting is very important for heuristic algorithm. For different optimisation problem, the parameters setting should be adjusted. The HCPSO provides the principles of selecting the parameters and for other approaches reader should refer to related literature. In summary, the improvement of running time via different parameter settings is a good way for finding the good solutions.

The link of this system and NAPA software is MACRO language. The MACRO language is the bridge and calls every calculation model. The output media of this bridge is text and text is also the input file media of the system in which the text file is the window of the application. Most results are shown in text files and therefore if the designers want to check the running process, checking the text file is a good choice.

9.5 Future Work

The learning based decision design support system successfully solved the experience sharing problem in the design support area. The following aspects need further study in order to improve the learning ability together with increasing the efficiency of design decision support system.

a. The linguistic attribute operations in SDLL need further study for finding a more effective approach for better classifying. In this study, the lazy method is employed to avoid useless treatments for the linguistic and leaves it to operate until the new case needs this information. If a more effective method can be found to treat the linguistic before it is used, it will further reduce the system’s run time.
b. The data format of SDLL can be updated. In SDLL, XML is accepted to represent the data. XML language is simple and easy to understand but it is not powerful enough to operate large scale data. Moreover, one of the advantages of XML is the online function, which is also the developing direction of this system. However in current research, this direction is not studied fully.
c. The systematic optimisation method should be developed to give more choices for the designers. In this study, the HCPSO, NSGAII is integrated into the system. This system also gives the interface to the third party software Model Frontier. But the detailed comparison of every approach from full ranges including initial value, parameter setting etc. should be processed to improve the calculation

ability of the system.

d. The conflict solving is an important research field from prior work to current research. This is also the important and very popular research area of multi-agent system. In this study, the simplified method was deployed and has been proven that it was effective. However for more complex society learning, the system method needs to be applied to this system. Currently, the research of society learning is at the starting stage and when new approaches are mature or developed, further study should be promoted to improve the performance of the system.
e. Q-learning approach in this study uses the look-up table which is effective for finite and discrete environment. When a more complex design environment is developed, a more effective approach should be utilised.
f. The examples used for HCPSO are unconstrained test functions. But there are always a large number of constraints in practical work of ship design. So the constraint test functions should be considered in future studies. When these test functions are tested, in addition to comparison of the generation numbers, the comparison should also include CPU time to demonstrate the high effectiveness of test functions.

More design problems should be tested in order to evaluate further this support system to extend the application ranges. More software such as ANSYS, CATIA should be employed to test the interface. As a design decision support system, it needs to be extended to different tasks to test the efficiency.

9.6 Conclusions

The aim of this research work is to develop a multi-agent ship design decision support system with learning ability, which can automatically improve the design according to the experience gained from the agents' self-determination for learning. There have been several conclusions derived from this research, however, the main conclusions emerging from this thesis are detailed as follows:

The HCPSO displays excellent performance with simple parameters setting, compared to other optimisation approaches such as NSGAII. For both generic optimisation functions and ship design applications, it improves significantly the accuracy as well as the computing time (run time)

However, these swarm optimisation and genetic algorithms are all sensitive to the parameters setting. When the population and generation in engineering optimisation are limited, the different parameters setting have a great influence on the final results. Therefore there is a huge risk by simply accepting random parameters without proper analysis The numerical analysis is necessary before engineering application.

When the learning function is combined with the optimization approach, it improves the quality of design as well as the design time. In some practical applications, it may have limitations on improving the quality of solutions further but the runtime will be reduced greatly. A significant part of the total run time is taken up by the third party software, in comparison to the optimisation process which only takes up a small part of the run time. Therefore in order to reduce the time, focus should be on reducing the call frequency of third party software in optimisation. In decision based ship design process, the Q-learning approach as reinforcement learning is very suitable for practical application. The integrated optimisation methods with the Q-learning module greatly improve the speed of convergence whilst the accuracy of the solution is maintained. The reduced time greatly improves the efficiency of the proposed system as well as the suitability for the practical design applications.

The independent agent structure of Q-learning increases the robustness of real world application and extends the adaptability of the methodology which means that this method can be applied properly to any optimisation approaches.

A multi-agent system can improve the efficiency of the system. The communication among the agents and environments is important to run the multi-agent system. In order to improve the system efficiency further, the conversation mechanism should be prioritized. The multi-agent system structure, adopted in this study, fits exactly to the nature of concurrent engineering on ship design. The multi-agent system better describes the design process and it is very easy to understand when this system is applied in practical design.

When the multi-agent system is utilised for parallel processing of ship design, it changes the traditional linear design to group design, which means every design process employs agents to operate different tasks. The intelligent environment takes charge of the communication among agents. This mode greatly improves the efficiency of the proposed system. What is more, the multi-agent structure makes the replacement of agents easier and the up-to-date knowledge can be used quickly.

In the decision making after optimization, the training samples are sensitive for machine learning methods. Before application, the training samples should be revised by the designers carefully. The machine learning based decision making solves the absence of experts using the previous experience to replace the human experts. This method successfully solves the problem of the final selection from Pareto solutions after optimisation. In traditional approaches; this selection is dependent on the choice of human experts each time, thus making the selection difficult. This improvement makes the proposed system suitable for fast design of ships. The designers can get the final solution directly and immediately without waiting for the evaluations from human experts.

Reference

Aamodt, A. (1994). A knowledge representation system for integration of general and case-specific knowledge. Proceedings from IEEE TAI-94, International conference on tools with artificial intelligence (1994). New Orleans.

Alkan, A. D. and K. Gülez (2004). "A knowledge-based computational design tool for determining preliminary stability particulars of naval ships." Naval Engineers Journal 116(4): 37-52.

Andrews, D. (1981). "Creative ship design." Transactions of RINA 123: 447-471.

Andrews, D. (1998). "A comprehensive methodology for the design of ships (and other complex systems)." Royal Society of London Proceedings 454(1968): 187.

Baddeley, A. D. (1986). Working memory, Oxford University Press.

Baumgartner, U., C. Magele, et al. (2004). "Optimality and particle swarm optimisation." IEEE Transactions on Magnetics 2004 40(2): 1172-1175.

Boulougouris, E. and A. Papanikolaou (2008). "Multi-objective optimisation of a float LNG terminal." Ocean engineering 35: 787-811.

Brown, A. and T. Mierzwicki (2004). "Risk metric for multi-objective design of naval ships." Naval Engineers Journal 116(2): 55-72.

Brown, A. and J. Salcedo (2003). "Multiple-objective optimisation in naval ship design." Naval Engineers Journal 115(4): 49-61.

Brown, A. and M. Thomas (1998). Reengineering the naval ship design process. Proceedings of from research to reality in ship systems engineering symposium. University of Essex, United Kingdom.

Buxton, I. L. (1972). "Engineering economics applied to ship design." Transactions of RINA 114: 409-428.

Caprace, J.-D., N. Losseau, et al. (2007). "A Data Mining Analysis Applied to a Straightening Process Database." Ship Technology Research 54(4): 7.

Chow, C. and H. Tsui (2004). Autonomous agent response learning by a multi-species particle swarm optimisation. In congress on evolutionary computation (CEC'2004). 1**:** 778-785.

Clerc, M. and J. Kennedy (2002). "The particle swarm-explosion, stability, and convergence in a multidimensional complex space." IEEE Transactions on evolutionary computation 6: 58-73.

Coello, C. and M. Lechuga (2002). MOPSO: A proposal for multiple objective particle swarm optimisation. In Congress on Evolutionary Computation (CEC'2002). 2**:** 1051- 1056.

Coello, C., G. Pulido, et al. (2004). "Handling multiple objectives with particle swarm optimisation." IEEE Transactions on Evolutionary Computation 8(3): 256-279.

Cui, H. and O. Turan (2009). "An improved PSO approach in a multi criteria decision making environment." Ship technology research 56(1): 13-23.

Deb, K. (2001). Multi-objective optimisation using evolutionary algorithms, JOHN WILEY & SONS Ltd.

Deb, K., L. Thiele, et al. (2001). Scalable test problems for evolutionary multi-objective optimisation, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology.

Delatte, B. and A. Butler (2003). " An object-oriented model for conceptual ship design supporting case-based design." Marine Technology 40(3): 158-167.

Deng, N. Y. and Y. J. Tian (2004). A new method in data mining: support vector machine. Beijing, Science Press.

Eamon, C. D. and M. Rais-Rohani (2009). "Integrated reliability and sizing optimization of a large composite structure." Marine Structures 22: 315-334.

Eberhart, R. and Y. Shi (2000). Comparing inertia weights and constriction factors in particle swarm optimization. Proceedings of the 2000 congress on evolutionary computation, 2000**:** 84-88.

Evans, J. H. (1959). "Basic design concepts." American Society of naval Engineers Journal: 671-678.

Fieldsend, J. and S. Singh (2002). A multiobjective algorithm based upon particle swarm optimisation, an efficient data structure and turbulence. In Proceedings of the 2002 U.K. Workshop on Computational Intelligence. Birmingham.

Fujita, K. and S. Akagi (1999). "Agent-based distributed design system architecture for basic ship design " Concurrent engineering Vol. 7(No. 2): 83-93

Harmon, M. E. and S. S. Harmon (1996). Reinforcement learning: a tutorial.

Holt, C. and A. Roth (2004). The nash equilibrium: a perspective. PNAS 2004**:** 3999-4002.

Hu, X. and R. Eberhart (2002). Multiobjective optimisation using dynamic neighborhood particle swarm optimisation. In Congress on Evolutionary Computation (CEC'2002). 2**:** 1677-1681.

Jang, B., D. Ko, et al. (2009). "Adaptive approximation in multi-objective optimization for full stochastic fatigue design problem." Marine structures 22: 610-632.

Jin, Y., M. Olhofer, et al. (2001). Dynamic weighted aggregation for evolutionary multi-objective optimisation: why does it work and how? Proceedings of the Genetic and Evolutionary Computation Conference. San Francisco.

Kaelbling, L. and M. Littman (1996). "Reinforcement learning: a survey." Journal of Artificial Intelligence Research 4: 237-285.

Kantardzic, M. (2003). Data mining—concepts, models, methods, and algorithms.

Kennedy, J. and R. Eberhart (1995). Particle swarm optimisation. Proc. of the IEEE Conf. on Neural Networks. Perth.

Klanac, A., S. Ehlers, et al. (2009). "Optimization of crashworthy marine structures." Marine structures 22: 670-690.

Klanac, A. and J. Jelovica (2009). "Vectorization and constraint grouping to enhance optimization of marine structures." Marine structures 22: 225-245.

Kowalskia, Z., M. Meler-Kapciab, et al. (2005). "CBR methodology application in an expert system for aided design ship's engine room automation " Expert Systems with Applications 29(2): 256-263.

Lee, D. (1997). "Multiobjective design of a marine vehicle with aid of design knowledge." International journal for numerical methods in engineering 40: 2665-2677.

Lee, K. H. and K.-y. Lee (2002). "Agent-based collaborative design system and conflict resolution based on a case-based reasoning approach." Artificial Intelligence for Engineering Design, Analysis and Manufacturing Volume 16 (Issue 2): 93-102

Lee, K. H., J. Oh, et al. (2006). Development of data miner for the ship design based on polynomial genetic programming. Lecture Notes in Computer Science, Springer Berlin / Heidelberg. Volume 4304/2006.

Li, X. (2003). A non-dominated sorting particle swarm optimizer for multiobjective optimisation. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2003), Lecture Notes in Computer Science 2723**:** 37-48.

Lim, T.-S., W.-Y. Loh, et al. (2000). "A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms." Machine Learning 40(3): 203-229.

Margarita, R. and C. Coello (2006). "Multi-objective particle swarm optimizers: a survey of the state-of-the-art." International Journal of Computational Intelligence Research 2: 287-308.

Mistree, F., W. F. Smith, et al. (1990). "Decision-based design: a contemporary paradigm for ship design." Transactions, Society of Naval Architects and Marine Engineers 98.

Moore, J. and R. Chapman (1999). Application of particle swarm to multiobjective optimisation, Department of Computer Science and Software Engineering, Auburn University.

Ölçer, A. (2008). "A hybrid approach for multi-objective combinatorial optimisation problems in ship design and shipping." Computers & OR 35(9): 2760-2775.

Ölçer, A., C. Tuzcu, et al. (2003). Internal hull subdivision optimisation of Ro-Ro vessels in multiple criteria decision making environment. Proceedings of the 8th International Marine Design Conference**:** 339-351.

Ölçer, A., C. Tuzcu, et al. (2006). "An integrated multi-objective optimisation and fuzzy multi-attributive group decision-making technique for subdivision arrangement of Ro-Ro vessels." Applied Soft Computing 6: 221-243.

Ölçer, A. I. (2001). Development of a new fuzzy multiple attribute decision making approach and its application to decision making in ship design and shipbuilding. Institute of Science and Technology. Istanbul, Turkey, Istanbul Technical University. PhD.

Ölçer, A. İ., C. Tuzcu, et al. (2005). "An integrated multi-objective optimisation and fuzzy multi-attributive group decision-making technique for subdivision arrangement of Ro-Ro vessels " Applied Soft Computing 6(3): 221-243.

Papanikolaou, A. (2009). Risk-based ship design methods, tools and applications, Springer.

Parsons, M. G., D. J. Singer, et al. (1999). A hybrid agent approach for set-based conceptual ship design. Proceedings of the 10th International Conference on Computer Applications in Shipbuilding, Cambridge. MA,USA.

Parsopoulos, K., D. Tasoulis, et al. (2004). Multiobjective optimisation using parallel vector evaluated particle swarm optimisation. In Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2004). 2**:** 823-828.

Parsopoulos, K. and M. Vrahatis (2002). Particle swarm optimisation method in multiobjective problems. In Proceedings of the 2002 ACM Symposium on Applied Computing (SAC'2002)**:** 603-607.

Peri, D. and E. Campana (2003). "Multidisciplinary design optimisation of a naval surface combatant." Journal of Ship Research 47(1): 1-12.

Peri, D. and E. Campana (2005). "High-Fidelity models and multiobjective global optimisation algorithms in simulation-based design." Journal of Ship Research 49(3): 159-175.

Pinto, A., D. Peri, et al. (2007). "Multiobjective optimisation of a containership using deterministic Particle Swarm Optimisation." Journal of Ship Research 51(3): 217-228.

Quinlan, J. R. (1986). "Induction of decision trees." Machine Learning: 81-106.

Quinlan, J. R. (1993). C4.5: programs for machine learning. San Francisco, CA, USA Morgan Kaufmann Publishers Inc.

Ray, T. and K. Liew (2002). "A swarm metaphor for multiobjective design optimisation." Engineering Optimisation 34(2): 141-153.

Richir, T., J. D. Caprace, et al. (2007). Multicriterion scantling optimization of the midship section of a passenger vessel considering IACS requirements. 10th international Symposium on Practical Design of Ships and Other Floating Structure. Houston.

Rigo, P. (2003). "An integrated software for scantling optimization and least production cost." Ship Technology Research 50(3): 135-140.

Russell, S. J. and P. Norvig (2003). Artificial intelligence: a modern approach, Prentice Hall, Englewood Cliffs, NJ, 2nd edition.

Safavian, S. and D. Landgrebe (1991). "A survey of decision tree classifier methodology." IEEE Transactions on Systems, Man, and Cybernetics 21(3): 660-674.

Sefrioui, M. and J. Periaux (2000). Nash genetic algorithms : examples and applications. In proceeding of the congress on evolutionary computation 2000**:** 509-516.

Sekulski, Z. (2009). "Least-weight topology and size optimization of high speed vehicle-passenger catamaran structure by genetic algorithm." Marine Structures 22: 691-711.

Sen, P. and M. Gerigk (1992). Some aspects of a knowledge-based expert system for preliminary ship subdivision design for safety. International Symposium PRADS'92. 2**:** 1187-1197.

Sen, P. and J.-B. Yang (1998). Multiple criteria decision support in engineering design Springer-Verlag London Limited

Shi, Y. and R. Eberhart (1999). Empirical study of particle swarm optimization. Proc of the Congress on Evolutionary Computation**:** 1945-1950.

Simon, A. H. (1983). "Search and reasoning in problem solving." Artificial Intelligence 21(1-2): 9-29.

Singer, D. J. and M. G. Parsons (2003). Evaluation of the effectiveness of a fuzzy logic software agent to aid design team negotiation and communication. 2nd International EuroConference on Computer and IT Application in the Martime Industries (COMPIT'03). Hamburg,Germany 220-234.

Srdoč, A., I. Bratko, et al. (2007). "Machine learning applied to quality management-a study in ship repair domain." Computers in Industry 58 (5): 464-473

Stopford, M. (1997). Maritime Economics. London, Routledge.

The UNCTAD secretariat, U. N. (2008). Review of maritime transport 2008. New York and Geneva.

Thomas, M. (1998). A pareto frontier for full stern submarines via genetic algorithm. Ocean Engineering Department. Cambridge, Massachusetts, USA, Massachusetts Institute of Technology. PhD.

Todd, D. and P. Sen (1997). A multiple criteria genetic algorithm for containership Loading. Proceedings of the Seventh International Conference on Genetic Algorithms**:** 674-681.

Todd, D. and P. Sen (1997). Multiple criteria scheduling using genetic algorithms in a shipyard environment. In Proceedings of the 9th International Conference on Computer Applications in Shipbuilding.

Todd, D. and P. Sen (1998). Tackling complex job shop problems using operation based scheduling. The Integration of Evolutionary and Adaptive Computing Technologies with Product/System Design and Realisation.

Trelea, I. C. (2003). "The particle swarm optimisation algorithm: convergence analysis and parameter selection." Information Processing Letters 85(6): 317-325.

Turan, O., B. Turkmen, et al. (2004). Case-based reasoning approach to internal hull subdivision design. 4th International Conference on Advanced Engineering Design. Glasgow.

Turkmen, B. (2005). A multi-agent system based conceptual ship design decision support system. Department of Naval Architecture and Marine Engineering, Universities of Glasgow and Strathclyde.

Turkmen, B. S. and O. Turan (2003). Multi-agent systems in ship design. Hamburg,Germany 2nd International EuroConference on Computer and IT Application in the Martime Industries (COMPIT'03) 459-471.

Turkmen, B. S. and O. Turan (2004). An application study of multi-agent systems in multi-criteria ship design optimisation. 3rd International EuroConference on Computer and IT Application in the Maritime Industries. Hamburg,Germany

Vapnik, V. N. (1995). The nature of statistical learning theory. New York, Springer-Verlag.

Vapnik, V. N. (1998). Statistical learning theory. New York, Wiely.

Vlassis, N. (2007). A concise introduction to multiagent systems and distributed artificial intelligence Morgan & Claypool.

Wang, L., Ed. (2005). Studies in fuzziness and soft computing. Support Vector Machines: Theory and Applications, Springer-Verlag GmbH.

Watkins, C. (1989). Learning from delayed rewards, University of Cambridge.

Watkins, C. and P. Dayan (1992). "Q-learning." Machine Learning 8(3-4): 279–292.

Zanic, V., J. Andric, et al. (2007). Decision support methodology for concept design of multi-deck ship structures. 10th international Symposium on Practical Design of Ships and Other Floating Structure. Houston.

Zaraphonitis, G., E. Boulougouris, et al. (2003). An integrated optimisation procedure for the design of RO-RO passenger ships of enhanced safety and efficiency. Proceedings of the 8th international marine design conference**:** 313-324.

Zheng, W. and H. Liu (2007). "A cooperative co-evolutionary and ε-dominance based MOPSO." Journal of Software 18: 109−119.

Zitzler, E., K. Deb, et al. (2000). "Comparison of multiobjective evolutionary algorithms: empirical results." Evolutionary Computation 8(2): 173-195.

Appendix A

Case Study for SDLL

A Ropax ship design is selected to evaluate this new integrated learning approach. The aim of this case study is to solve the problem of selection of attributes in ship design, as this is a common problem in concept design. In most situations, the ship owner provides several basic requirements and the naval architects should provide the range of detailed attributes according to previous design cases. The difficulty of this design problem is that naval architects have to use the past experience to select the best design case on the balance of other attributes.

The operation of this case study is organized according the process which is presented in Figure A.1.

Pre-treatment Stage

The aim of this stage is to introduce the previous design instances, as in this step, these design instances are simply stored in the system without any category.

There are fourteen Ropax ships with six attributes including five numerical attributes and one linguistic attribute as shown in Table A.1. The requirement of ship owner is to design a new Ropax ship with speed higher than 20 knots and very good cargo capacity. So the designers and design decision support system should select the ranges of other attributes.

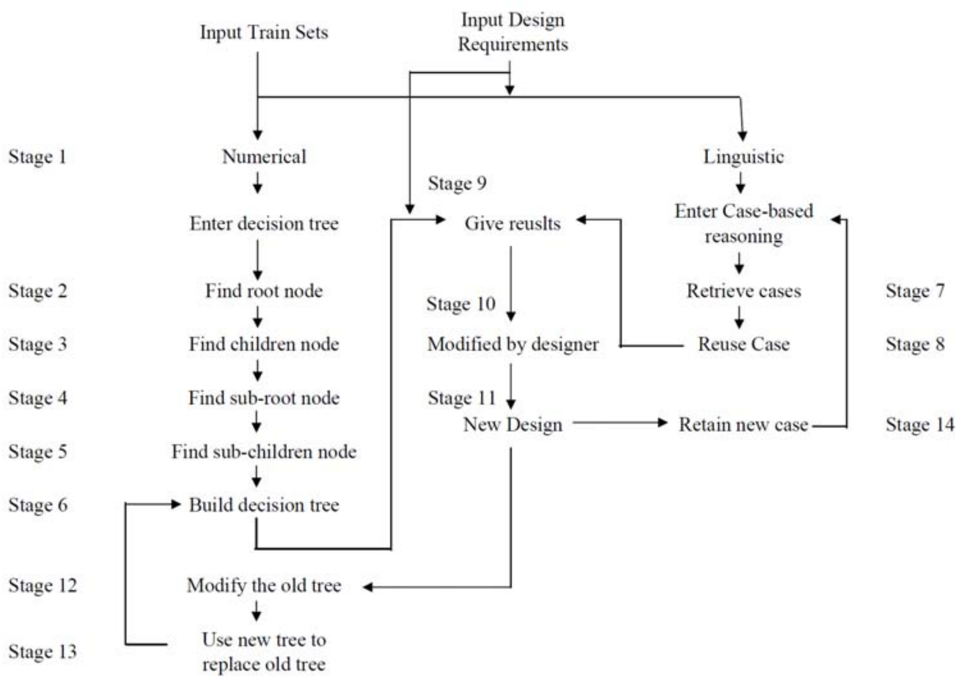


Figure A.1 Workflow of case study of Ropax ship

Stage 1 distinguish numerical or linguistic

The aim of stage 1 is to distinguish the different categories (numerical or linguistic) of six attributes.

From Table A.1, Length, Breadth, Design Draft, Deadweight and Speed are numerical attributes and these attributes will be entered into the decision tree part. The ability of Capacity belongs to linguistic form and will be directly entered into the data warehouse which means that it will not be operated until application stages.

Stage 2 Find root node

The aim of stage 2 is to find the root node for the whole decision tree, which is the most important step for building this decision tree.

In order to find the root node, a classification is necessary to tell the design decision support system which instances are good and which instances are bad for this situation. The training examples can be classified into two groups according to the speed (20 knots):

Less than 20 knots (8): No 1, 2, 4, 5, 6, 9, 10, 13，

More than 20 knots (6): No 3, 7, 8, 11, 12, 14

Because the new design requires that the speed is higher than 20 knots, this study will define the cases with more than 20 knots as YES and others as NO. So there are eight training examples belonging to YES and six training examples belonging to NO. According to definition and explanation given in section 4.4.2, Equation (4.1) can be rewritten as Equation (A.1):

$$Entropy(S) = -\frac{p_+}{p_+ + p_-}\log_2\left(\frac{p_+}{p_+ + p_-}\right) - \frac{p_-}{p_+ + p_-}\log_2\left(\frac{p_-}{p_+ + p_-}\right) \qquad \text{(A.1)}$$

In this example:

$$\begin{aligned} Entropy(S) &= -(6/14)\log_2(6/14) - (8/14)\log_2(8/14) \\ &= 0.9852 \end{aligned}$$

Table A. 1 The training sets of fourteen ROPAX ships

Att.	Length m	Breadth m	Design Draft m	Deadweight t	Speed knots	The ability of Capacity
Type of Att.	numerical	numerical	numerical	numerical	numerical	linguistic
Case No						
1	169.00	25.60	6.70	11843	14.60	good
2	150.43	23.40	5.60	6200	19.20	normal
3	166.75	23.40	5.80	6170	22.00	normal
4	183.00	28.70	6.80	9005	18.70	very good
5	126.40	21.00	6.00	5238	18.00	normal
6	142.50	23.20	5.00	4888	18.00	normal
7	264.60	32.26	10.70	39087	20.60	good
8	197.00	25.90	7.00	9000	21.10	very good
9	141.26	21.00	6.00	4500	19.20	excellent
10	183.40	25.20	7.50	12500	18.00	normal
11	152.00	23.60	6.30	7200	20.00	normal
12	157.96	25.20	6.50	7666	22.30	normal
13	193.00	26.00	6.60	10090	18.00	normal
14	255.72	35.97	8.99	21133	24.00	excellent

Then, the gain of every attribute will be calculated for analysis. In order to determine the gain, the intermediate value is calculated. The Table A.2 shows the intermediate value of every attributes.

Table A. 2 The intermediate value and calculation point of selected attributes

Att.	Length m	Breadth m	Design Draft m	Deadweight t
Type of Att.	numerical	numerical	numerical	numerical
The calculation point	195	25	6.5	8000

In Table A.2, the calculation point is the dividing point to separate this attribute into different range. This point can be decided by the designers according to rules, regulations and experience. For example, in this study, the calculation point is decided by experience on Ropax ships.

1. Length (m)

Table A.3 Gain of length attribute in level 1

Category \ Length		I	II	III
		L1 (<195)	L2 (≥195)	L
A	Yes (Speed≥20)	3	3	6
B	No (Speed<20)	8	0	8
C	Sum	11	3	14

The Table A.3 shows the gain value of length attribute in the root node calculation. The column I is the amount of ships with the length less than 195 m and column II is the ship with the length more than 195 m. The 'AI' is the ship cases with speed more than 20 knots in column I and 'BI' is the ship cases with speed less than 20 knots in column I. The 'AII' is the ship cases with speed more than 20 knots in column II and 'BII' is the ship cases with speed less than 20 knots in column II. The 'AI' and 'BI' are used to calculate $Entropy(S_{<195})$ according to Equation A.1, which is the basic part for Gain calculation. The 'AII' and 'BII' are used to calculate $Entropy(S_{\geq 195})$.

$$
\begin{aligned}
Gain(S, Length) &= Entropy(S) - \sum_{v \in Values(L<195, L\geq 195)} \frac{|Sv|}{|S|} Entropy(Sv) \\
&= Entropy(S) - (\frac{11}{14} Entropy(S_{<195}) + \frac{3}{14} Entropy(S_{\geq 195}))
\end{aligned}
$$

$$
\begin{aligned}
Entropy(S_{<195}) &= -(3/11)\log_2(3/11) - (8/11)\log_2(8/11) \\
&= 0.8454
\end{aligned}
$$

$$
\begin{aligned}
Entropy(S_{\geq 195}) &= -(3/3)\log_2(3/3) - (0/3)\log_2(0/3) \\
&= 0
\end{aligned}
$$

So Gain can be obtained from Equation 4.2:

$$Gain(S, Length) = Entropy(S) - (\frac{11}{14} Entropy(S_{<195}) + \frac{3}{14} Entropy(S_{\geq 195}))$$
$$= 0.9852 - (\frac{11}{14} \times 0.8454) - (\frac{3}{14} \times 0)$$
$$= 0.3210$$

The Max-Gain has been explained in section 4.4. According to the explanation, this gain means the smallest expected size of the sub-trees of this node. Here the gain is calculated on the information theory and the aim of this calculation is to compare with other gains of the attributes to decide the root node.

2. Breadth (m)

Table A.4 Gain of Breadth attribute in level 1

	B1 (<25)	B2 (⩾25)	B
Yes (Speed⩾20)	2	4	6
No (Speed<20)	4	4	8
Sum	6	8	14

$$Entropy(S_{B<25}) = -(2/6)\log_2(2/6) - (4/6)\log_2(4/6)$$
$$= 0.9183$$

$$Entropy(S_{B\geq 25}) = -(4/8)\log_2(4/8) - (4/8)\log_2(4/8)$$
$$= 1$$

$$Gain(S, Breadth) = Entropy(S) - \sum_{v \in Values(B<25, B\geq 25)} \frac{|Sv|}{|S|} Entropy(Sv)$$
$$= Entropy(S) - (\frac{6}{14} Entropy(S_{B<25}) + \frac{8}{14} Entropy(S_{B\geq 25}))$$
$$= 0.9852 - (\frac{6}{14} \times 0.9183) - (\frac{8}{14} \times 1)$$
$$= 0.0202$$

3. Design Draft

Table A.5 Gain of Draft attribute in level 1

	D1 (<6.5)	D2 (≥6.5)	D
Yes (Speed≥20)	2	4	6
No (Speed<20)	4	4	8
Sum	6	8	14

$$Entropy(S_{D<6.5}) = -(2/6)\log_2(2/6) - (4/6)\log_2(4/6)$$
$$= 0.9183$$

$$Entropy(S_{D\geq 6.5}) = -(4/8)\log_2(4/8) - (4/8)\log_2(4/8)$$
$$= 1$$

$$Gain(S, \text{Design Draft}) = Entropy(S) - \sum_{v \in Values(D<6.5, D\geq 6.5)} \frac{|Sv|}{|S|} Entropy(Sv)$$
$$= Entropy(S) - (\frac{6}{14} Entropy(S_{D<6.5}) + \frac{8}{14} Entropy(S_{D\geq 6.5}))$$
$$= 0.9852 - (\frac{6}{14} \times 0.9183) - (\frac{8}{14} \times 1)$$
$$= 0.0202$$

4. Deadweight

Table A.6 Gain of Deadweight attribute in level 1

	DW1 (<8000)	DW2 (≥8000)	D
Yes (Speed≥20)	3	3	6
No (Speed<20)	4	4	8
Sum	7	7	14

$$Entropy(S_{DW<8000}) = -(3/7)\log_2(3/7) - (4/7)\log_2(4/7)$$
$$= 0.9852$$

$$Entropy(S_{DW\geq 8000}) = -(3/7)\log_2(3/7) - (4/7)\log_2(4/7)$$
$$= 0.9852$$

$$Gain(S, Deadweight) = Entropy(S) - \sum_{v \in Values(DW<8000, DW \geq 8000)} \frac{|Sv|}{|S|} Entropy(Sv)$$

$$= Entropy(S) - (\frac{7}{14} Entropy(S_{DW<8000}) + \frac{7}{14} Entropy(S_{DW \geq 8000}))$$

$$= 0.9852 - (\frac{7}{14} \times 0.9852) - (\frac{7}{14} \times 0.9852)$$

$$= 0$$

The gain of attribute is equal to zero. This means that this node has merely to divide the cases into tow parts equally. In other words, this node is not the good node to divide the cases.

All the gain values of attributes are collected in Table A.7.

Table A.7 Gain of attributes in level 1

No	Gain Name	Value
1	Length	0.3210
2	Breadth	0.0202
3	Design Draft	0.0202
4	Deadweight	0

According to the results on gain measure, the Length attribute has the max-gain, which means it provides the best prediction of the target attribute over the training examples. So Length is selected as the decision attribute for the root node. After the update, the new calculation will be performed to decide the sublevel.

Stage 3 Find children node

The aim of stage 3 is to make clear the children node of level 1. The level 1 means the root node level.

Because the Length has been selected as root node, only three node candidates are left: Breadth, Design Draft and Deadweight.

Stage 4 Find sub-root node

The aim of stage 4 is to find the second level root node.

Because Length is selected as the root node, the training cases will be divided into two parts the ships with length less than 195 m and the ships with length more than 195 m.

Part 1 The length less than 195 m (as shown in Table A.8)

Table A.8 The training set of the length less than 195 m in level 2

Att.	Length m	Breadth m	Deign Draft m	Deadweight t	Speed knots	The ability of Capacity
Type of Att.	numerical	numerical	numerical	numerical	numerical	linguistic
Case No						
1	169.00	25.60	6.70	11843	14.60	good
2	150.43	23.40	5.60	6200	19.20	normal
3	166.75	23.40	5.80	6170	22.00	normal
4	183.00	28.70	6.80	9005	18.70	very good
5	126.40	21.00	6.00	5238	18.00	normal
6	142.50	23.20	5.00	4888	18.00	normal
9	141.26	21.00	6.00	4500	19.20	excellent
10	183.40	25.20	7.50	12500	18.00	normal
11	152.00	23.60	6.30	7200	20.00	normal
12	157.96	25.20	6.50	7666	22.30	normal
13	193.00	26.00	6.60	10090	18.00	normal

Except for the attribute of Length, there are three attributes and they are calculated as follows.

In this part,

Ships with the speed less than 20 knots (8): No 1, 2, 4, 5, 6, 9, 10, 13

Ships with the speed more than 20 knots (3): No 3, 11, 12

$$Entropy(S) = -(8/11)\log_2(8/11) - (3/11)\log_2(3/11)$$
$$= 0.8454$$

1. Breadth (m)

Table A.9 Gain of Breadth attribute in level 2

	B1 (<25)	B2 (⩾25)	B
Yes (Speed⩾20)	2	1	3
No (Speed<20)	4	4	8
Sum	6	5	11

$$Entropy(S_{B<25}) = -(2/6)\log_2(2/6) - (4/6)\log_2(4/6)$$
$$= 0.9183$$

$$Entropy(S_{B\geq 25}) = -(1/5)\log_2(1/5) - (4/5)\log_2(4/5)$$
$$= 0.7219$$

$$Gain(S, Breadth) = Entropy(S) - \sum_{v \in Values(B<25, B\geq 25)} \frac{|Sv|}{|S|} Entropy(Sv)$$
$$= 0.8454 - (\frac{6}{11} \times 0.9183) - (\frac{5}{11} \times 0.7219)$$
$$= 0.0164$$

2. Design Draft

Table A.10 Gain of Draft attribute in level 2

	D1 (<6.5)	D2 (⩾6.5)	D
Yes (Speed⩾20)	2	1	3
No (Speed<20)	4	4	8
Sum	6	5	11

$$Entropy(S_{D<6.5}) = -(2/6)\log_2(2/6) - (4/6)\log_2(4/6)$$
$$= 0.9183$$
$$Entropy(S_{D\geq 6.5}) = -(1/5)\log_2(1/5) - (4/5)\log_2(4/5)$$
$$= 0.7219$$

$$Gain(S, \text{Design Draft}) = Entropy(S) - \sum_{v \in Values(D<6.5, D\geq 6.5)} \frac{|Sv|}{|S|} Entropy(Sv)$$

$$= Entropy(S) - (\frac{6}{11} Entropy(S_{D<6.5}) + \frac{5}{11} Entropy(S_{D\geq 6.5}))$$

$$= 0.8454 - (\frac{6}{11} \times 0.9183) - (\frac{5}{11} \times 0.7219)$$

$$= 0.0164$$

3. Deadweight

Table A.11 Gain of Deadweight attribute in level 2

	DW1 (<8000)	DW2 (⩾8000)	D
Yes (Speed⩾20)	3	0	3
No (Speed<20)	4	4	8
Sum	7	4	11

$$Entropy(S_{DW<8000}) = -(3/7)\log_2(3/7) - (4/7)\log_2(4/7)$$
$$= 0.9852$$

$$Entropy(S_{DW\geq 8000}) = -(0/4)\log_2(0/4) - (4/4)\log_2(4/4)$$
$$= 0$$

$$Gain(S, \text{Deadweight}) = Entropy(S) - \sum_{v \in Values(DW<8000, D\geq 8000)} \frac{|Sv|}{|S|} Entropy(Sv)$$

$$= Entropy(S) - (\frac{7}{11} Entropy(S_{DW<8000}) + \frac{4}{11} Entropy(S_{DW\geq 8000}))$$

$$= 0.8454 - (\frac{7}{11} \times 0.9852) - (\frac{4}{11} \times 0)$$

$$= 0.2185$$

Table A.12 Gain of attributes on the range of the length less than 195 m in lever 2

No	Gain Name	Value
1	Breadth	0.0164
2	Design Draft	0.0164
3	Deadweight	0.2185

According to the results on gain measure in Table A.12, the Deadweight attribute has max-gain. So, the Deadweight attribute is selected as one of the nodes of level 2.

Part 2 The length more than 195 m (as shown in Table A.13)

Table A.13 The training set of length more than 195 m in level 2

Att.	Length m	Breadth m	Deign Draft m	Lane length m	Deadweight t	Speed knots	The ability of Capacity
Type of Att.	numerical	numerical	numerical	numerical	numerical	numerical	linguistic
Case No							
7	264.60	32.26	10.70	6100	39087	20.60	good
8	197.00	25.90	7.00	2820	9000	21.10	very good
14	255.72	35.97	8.99	7200	21133	24.00	excellent

In this part, the speeds of all examples are more than 20 knots. So they will not be analysed. This means the length more than 195 corresponds to the speed more than 20 knots indicating that this node (length more than 195) can directly be used as sub-root-node. The ship with the length more than 195 has the character of speed over 20 knots.

So in the level 2, there is one node: Deadweight attribute for the length less than 195 m.

Stage 5 Find sub-children node

The aim of stage 5 is to find the left attributes for the next level. When Deadweight has been decided as root node of level 2 with Length is the root node of level 1, there are only two attributes being left: Breadth and design draft.

Return to Stage 4 Find sub-root node

The calculation will be continued to decide the nodes in level 3. So the calculation returns to stage 4. According to stage 5, there are two attributes: Design draft and Breadth which will be taken in to account.

Part 1 in level 3 (Deadweight less than 8000t and length less than 195 m)

Table A.14 The training set of Deadweight less than 8000t, Length less than 195 m in level 3

Att.	Length m	Breadth m	Deign Draft m	Deadweight t	Speed knots	The ability of Capacity
Type of Att.	numerical	numerical	numerical	numerical	numerical	linguistic
Case No						
2	150.43	23.40	5.60	6200	19.20	normal
3	166.75	23.40	5.80	6170	22.00	normal
5	126.40	21.00	6.00	5238	18.00	normal
6	142.50	23.20	5.00	4888	18.00	normal
9	141.26	21.00	6.00	4500	19.20	excellent
11	152.00	23.60	6.30	7200	20.00	normal
12	157.96	25.20	6.50	7666	22.30	normal

In this part,

Ships with the speed less than 20 knots (5): No 2, 5, 6, 9, 11

Ships with the speed more than 20 knots (2): No 3, 12

$$\begin{aligned} Entropy(S) &= -(5/7)\log_2(5/7) - (2/7)\log_2(2/7) \\ &= 0.8631 \end{aligned}$$

2. Breadth (m)

Table A.15 Gain of Breadth attribute in level 2

	B1 (<25)	B2 (⩾25)	B
Yes (Speed⩾20)	2	1	3
No (Speed<20)	4	0	4
Sum	6	1	7

$$Entropy(S_{B<25}) = -(2/6)\log_2(2/6) - (4/6)\log_2(4/6)$$
$$= 0.9183$$

$$Entropy(S_{B\geq 25}) = -(1/1)\log_2(1/1) - (0/1)\log_2(0/1)$$
$$= 0$$

$$Gain(S, Breadth) = Entropy(S) - \sum_{v \in Values(B<25, B\geq 25)} \frac{|Sv|}{|S|} Entropy(Sv)$$
$$= 0.8631 - (\frac{6}{7} \times 0.9183) - (\frac{1}{7} \times 0)$$
$$= 0.0760$$

2. Design Draft

Table A.16 Gain of Draft attribute in level 2

	D1 (<6.5)	D2 (⩾6.5)	D
Yes (Speed⩾20)	2	1	3
No (Speed<20)	4	0	4
Sum	6	1	7

$$Entropy(S_{D<6.5}) = -(2/6)\log_2(2/6) - (4/6)\log_2(4/6)$$
$$= 0.9183$$

$$Entropy(S_{D\geq 6.5}) = -(1/1)\log_2(1/1) - (0/1)\log_2(0/1)$$
$$= 0$$

$$Gain(S, \text{Design Draft}) = Entropy(S) - \sum_{v \in Values(D<6.5, D\geq 6.5)} \frac{|Sv|}{|S|} Entropy(Sv)$$
$$= Entropy(S) - (\frac{6}{7} Entropy(S_{D<6.5}) + \frac{1}{7} Entropy(S_{D\geq 6.5}))$$
$$= 0.8631 - (\frac{6}{7} \times 0.9183) - (\frac{1}{7} \times 0)$$
$$= 0.0760$$

Table A.17 Gain of attributes on the range of Deadweight less than 8000t, Length less than 195 m in level 3

No	Gain Name	Value
1	Breadth	0.0760
2	Design Draft	0.0760

From the Table A.17, it can be seen that the gains of Breadth and Design Draft are the same. This means If any one of these two attributes can be selected as nodes, the size of sub-tree are same, in other words, these two attributes have no difference whether they are dominate or non-dominate.

The reason for this situation can be studied by analysing the data in Table A.14. In Table A.14, if design draft is more than or equal to 6.5 m (only instance 12 satisfies this condition in Table A.14), the speed is more than 20 knots (the speed of instance 12 is more than 20knots) and if breadth is more than 25 m (still only instance 12 satisfies this condition in Table A.14), the speed is also more than 20 knots. So the instance 12 is the different instance for other instances. Other instances can not be classified according to current information. In the programming of proposed system, if all gains are the same, the system will randomly select one of the attributes as root node. Here, in this case, the breadth is selected as node. The instances with Breadth attribute of less than 25 m are classified as speed less than 20.

But from Table A.14, there are two instances (instance 3 and 11) which are not correct according to this classification. In an ideal state, all the instances can be correctly classified using calculation point. But in practice, there are error points which can not be correctly classified in decision tree. The reason is that the calculation point is decided by experience or rules and it sometimes can not be correct for all design situations. If the system thinks these errors are big enough to effect the classification, it will automatically revise the dividing point. In this case, two error instances can be accepted. So it is normal to have error points and the decision tree will be modified in the following run which will revise these error points.

Part 2 in level 3 (Deadweight more than 8000t and length less than 195 m)

Table A.18 The training set of Deadweight more than 8000 t,

Length less than 195 m in lever 3

Att.	Length m	Breadth m	Deign Draft m	Deadweight t	Speed knots	The ability of Capacity
Type of Att.	numerical	numerical	numerical	numerical	numerical	linguistic
Case No						
1	169.00	25.60	6.70	11843	14.60	good
4	183.00	28.70	6.80	9005	18.70	very good
10	183.40	25.20	7.50	12500	18.00	normal
13	193.00	26.00	6.60	10090	18.00	normal

As it can be seen in Table A.18, the speeds of all the examples are less than 20 knots. So there is no need for further analysis.

Stage 6 Build decision tree

The Figure A.2 is the final decision tree according to above analysis.

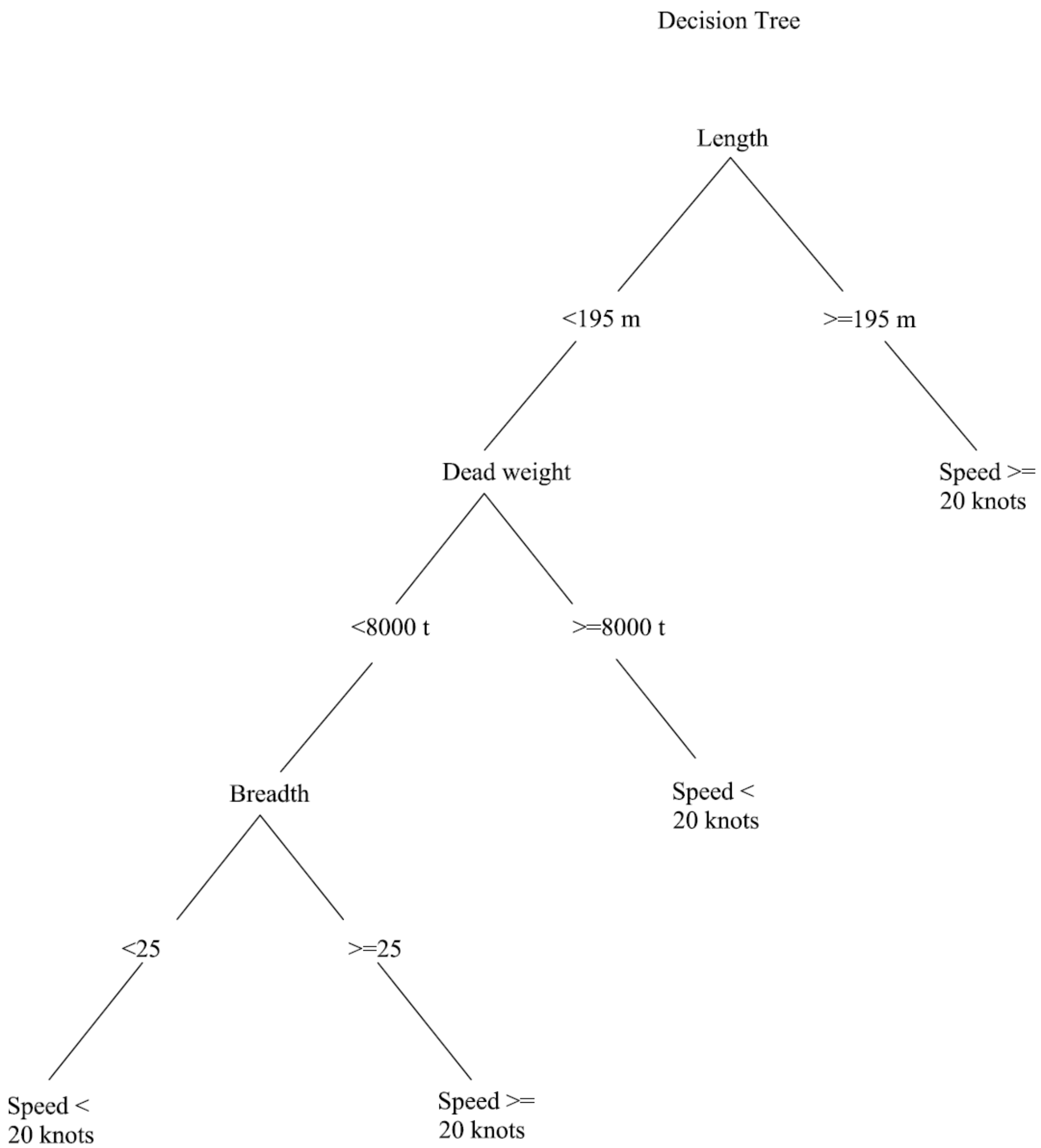


Figure A. 2 The final decision tree

The Figure A.2 displays the whole decision tree in this case. In the practical ship design process, the experience is important for the creative objective. For example, in the optimisation process of ship design, the boundaries and steps are very important for optimisation. If the wrong or unsuitable boundaries and steps are used, the quality of the optimisation will be greatly reduced. Normally, the optimisation will select rules as the boundaries but this selection is fuzzy measurement and for some design variables, there are no clear definition in rules and regulations. Therefore, how to use the experience to assist the range of design variables has to become a critical problem. Over wide range will effect the feasible of design when narrow range will limit the optimisation results. The designers have to check many of previous cases to obtain experience.

This SDLL provide a good selection for the above situation. From Figure A.2, if the designers want to design a new ship which speed is more than 20 knots, they can select the length more than 195m or the length less than 195m but breadth more than or equal to 25 m. When the length is limited to less than 195m, the breadth will be more than or equal to 25 m.

The design aim is a new ROPAX ship with speed higher than 20 knots and very good cargo capacity. According to Figure A.2, there are two ways to select the design. One is the length more than 195 m. The other is the length less than 195 m but deadweight less than 8000 t and breadth more than 25 m. In this study, the designer selects length more than 195m (calculation point). The corresponding instances are listed in Table A.19.

Stage 7 Retrieve cases

The aim of stage 7 is to find previous instances which satisfy the selection standard according to the decision tree built in stage 6.

The requirement of very good cargo capacity is taken into account and CBR method used to select solutions from Table A.19.

Table A.19 The examples with length than 195m

Att.	Length m	Breadth m	Deign Draft m	Deadweight t	Speed knots	The ability of Capacity
Type of Att.	numerical	numerical	numerical	numerical	numerical	linguistic
Case No						
7	264.60	32.26	10.70	39087	20.60	good
8	197.00	25.90	7.00	9000	21.10	very good
14	255.72	35.97	8.99	21133	24.00	excellent

Stage 8 Reuse cases

The aim of stage 8 is to use the instances in Table A.19 to provide a new design.

In this study, simple relationship between fuzzy numbers and linguistic attributes are provided as shown in Table A.20:

Table A.20 The relationship of linguistic attributes and fuzzy numbers

linguistic	Bad	Normal	Good	Very Good	Excellent
Fuzzy number	1	2	3	4	5

The distance of linguistic attributes via Euclidean Distance can be calculated as:

$$d(x,y) = \sqrt{\sum_A w^A (x^A - y^A)^2}$$

x_A is the value of attribute A for example x;

w_A is a nonnegative real valued parameter that specifies the relative weight of attribute A;

In this calculation, $w_A = 1$;

$d(7,y) = \sqrt{(3-4)^2} = 1$;

$d(8,y) = \sqrt{(4-4)^2} = 0$;

$d(14, y) = \sqrt{(5-4)^2} = 1;$

Stage 9 Give the result

From the above calculation, $d(8, y) < d(7, y) = d(14, y)$, instance 8 is the best instance for this design and recommended to the designer.

Table A.21 The recommended instance for new design

Att.	Length m	Breadth m	Design Draft m	Deadweight t	Speed knots	The ability of Capacity
Type of Att.	numerical	numerical	numerical	numerical	numerical	linguistic
8	197.00	25.90	7.00	9000	21.10	very good

Stage 10 Modified by designer

The aim of stage 10 is to define the final design by introducing the revision of designers. The design decision support system provides instance 8 as reference case. This stage is processed by designers

Stage 11 New design

The final new design can be designed according to Figure A.2 and Table A.21.

Stage 12 Modify the old tree and Stage 13 use new tree to replace old tree

The decision tree can be revised according to new design.

Stage 14 retain new case

The new design will be stored in Table A.1 as No 15 for next design.

Appendix B

Loads Calculation of CSR

The loads in this case study are selected according to the CSR of IACS and the original design data. If the original design data provides the detailed values of loads, the case will use these values, and if not, the case will calculate according to CSR. For reducing the complexity of the calculation, only case H1 of CSR is considered in this case study. For weight control, the loading is corresponded to probability level 10^{-4} and for fatigue coefficient, it is 10^{-8}.

B.1 Hull girder loads

Still water bending moment

Still water bending moment calculated via CSR:

$M_{SW,H}(CSR) = 1006547\ kN \cdot m$

$M_{SW,S}(CSR) = 924006\ kN \cdot m$

Still water bending moment calculated in design proposal:

$M_{SW,H}(design) = 1700000\, kN \cdot m$

$M_{SW,S}(design) = 1500000\, kN \cdot m$

The still water bending moment should be bigger one, so it will be the values in design proposal.

$M_{SW,H} = M_{SW,H}(design) = 1700000\, kN \cdot m$

$M_{SW,S} = M_{SW,S}(design) = 1500000\, kN \cdot m$

Still water shear force

The still water shear force is provided by the design proposal:

$Q_{SW}(+/-) = 8.5920 \times 10^4 \ kN$

Wave loads

Vertical wave bending moments

The vertical wave bending moments in intact condition are calculated via CSR:

$M_{WV,H} = 1481161 \ kN \cdot m$

$M_{WV,S} = 1563702 \ kN \cdot m$

Vertical wave shear force

$Q_{WV}(+/-) = 16585 \ kN$

Horizontal wave bending moment

$M_{WH} = 1124902 \ kN \cdot m$

Wave torsional moment

$M_{WT} = 285838 \ kN \cdot m$

B.2 External pressures

Hydrostatic pressure

$$p_s = \begin{cases} 10045 \times (11.1 - z) & z \le 11.1 \\ 0 & z > 11.1 \end{cases}$$

In this study, ABQUAS is employed to simulate the loading case. Hydrostatic pressure can be directly realized in ABQUAS, but A FORTAN coding is created for assist to simulate for combining with other loads.

Hydrodynamic pressures

For Load case H1:

$p_{H1} = -k_l k_p p_{HF}$

In this case:

$$\begin{cases} k_l = 1 + \dfrac{240}{17}(1 - \sqrt{\dfrac{|y|}{15}})\left|\dfrac{x}{180} - 0.5\right|^3 & for \quad 0.0 \le x/180 \le 0.5 \\ k_l = 1 + \dfrac{120}{17}(3 - \sqrt{\dfrac{|2y|}{15}})\left|\dfrac{x}{180} - 0.5\right|^3 & for \quad 0.5 \le x/180 \le 1.0 \end{cases}$$

$$k_p = 0.37\cos\left(\frac{2\pi|x-90|}{180}\right) - 0.63 \qquad \begin{array}{l} f_{nl} = 0.9 \quad \text{for the probability level } 10^{-8} \\ f_{nl} = 1.0 \quad \text{for the probability level } 10^{-4} \end{array}$$

$$\lambda_{H1} = 0.6\left(1 + \frac{T_{LC}}{T_S}\right)L = 203.04$$

So

$$P_{HF} = 3 f_p f_{nl} C\sqrt{\frac{L+\lambda-125}{L}}\left(\frac{z}{T_{LCi}} + \frac{|2y|}{B_i} + 1\right)$$

Because this section is the midship, the T_{LCi} and B_i are the midship value.

$$\begin{cases} f_p = 1.0 \quad \text{for the probability level } 10^{-8} \\ f_p = 0.5 \quad \text{for the probability level } 10^{-4} \end{cases}$$

$$\begin{cases} P_{HF} = 30.5022\left(\dfrac{z}{11.1} + \dfrac{|2y|}{30} + 1\right) & \text{for the probability level } 10^{-8} \\ P_{HF} = 16.9457\left(\dfrac{z}{11.1} + \dfrac{|2y|}{30} + 1\right) & \text{for the probability level } 10^{-4} \end{cases}$$

So

For for the probability level 10^{-8} and $for \quad 0.0 \le x/180 \le 0.5$,

$$p_{H1} = -(1 + \frac{240}{17}(1 - \sqrt{\frac{|y|}{15}})\left|\frac{x}{180} - 0.5\right|^3) \times (0.37\cos\left(\frac{2\pi|x-90|}{180}\right) - 0.63) \times 30.5022\left(\frac{z}{11.1} + \frac{|2y|}{30} + 1\right)$$

for the probability level 10^{-8} and $for \quad 0.5 \le x/180 \le 1.0$

$$p_{H1} = -(1 + \frac{120}{17}(3 - \sqrt{\frac{|2y|}{15}})\left|\frac{x}{180} - 0.5\right|^3) \times (0.37\cos\left(\frac{2\pi|x-90|}{180}\right) - 0.63) \times 30.5022\left(\frac{z}{11.1} + \frac{|2y|}{30} + 1\right)$$

for the probability level 10^{-4} and $for \quad 0.0 \le x/180 \le 0.5$

$$p_{H1} = -(1 + \frac{240}{17}(1 - \sqrt{\frac{|y|}{15}})\left|\frac{x}{180} - 0.5\right|^3) \times (0.37\cos\left(\frac{2\pi|x-90|}{180}\right) - 0.63) \times 16.9457\left(\frac{z}{11.1} + \frac{|2y|}{30} + 1\right)$$

for the probability level 10^{-4} and $for \ \ 0.5 \le x/180 \le 1.0$

$$p_{H1} = -(1+\frac{120}{17}(3-\sqrt{\frac{|2y|}{15}})\left|\frac{x}{180}-0.5\right|^3)\times(0.37\cos\left(\frac{2\pi|x-90|}{180}\right)-0.63)\times 16.9457\left(\frac{z}{11.1}+\frac{|2y|}{30}+1\right)$$

In this part, the detailed calculation of other case loads is processed Here, the results will be provided.

Correction to hydrodynamic pressures

For the positive hydrodynamic pressure at the waterline

$$\begin{cases} p_{W,C} = p_{W,WL} + \rho g(T_{LCi} - z), & T_{LCi} \le z \le h_W + T_{LCi} \\ p_{W,C} = 0 & z \ge h_W + T_{LCi} \end{cases}$$

Where，

$$h_W = \frac{p_{W,WL}}{\rho g}$$

For the negative hydrodynamic pressure at the waterline

$p_{W,C} = p_W$ without being taken less than $\rho g(z - T_{LCi})$

B.3 Internal pressures and forces

1. Dry bulk cargo pressure in still water

Step 1 Calculate h_c

$h_c = 12.5$ m

l_H : Length, in m, of the compartment

$p_{CS} = \rho_C g K_C (h_C + h_{DB} - z)$

$\rho_C = 1$ t/m^3

$g = 9.81$ m/s^2

$h_{DB} = 1.8$ m

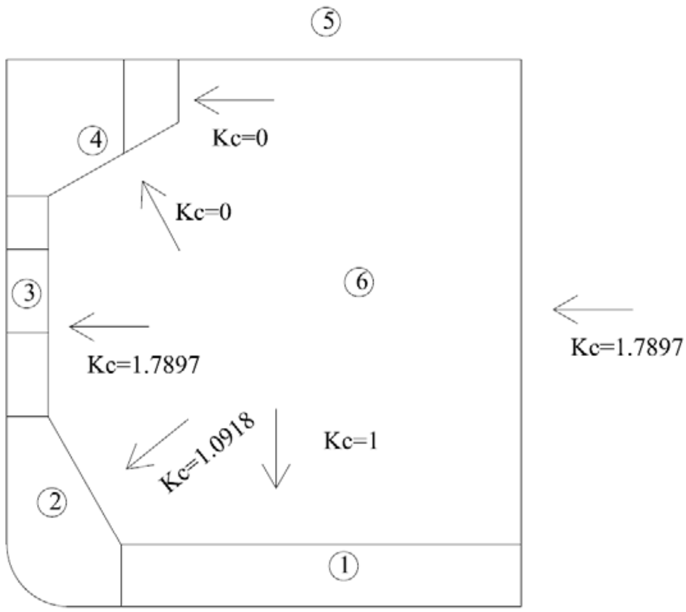


Figure B.1 The Kc values of Dry bulk cargo pressure in still water

Figure B.1 shows the Kc value of Dry bulk cargo pressure in still water.

So

$p_{CS} = 1 \times 9.81 \times K_C \times (14.3 - z);$

$p_{CS-1} = 9.81 \times (14.3 - z);$

$p_{CS-2} = 10.71 \times (14.3 - z);$

$p_{CS-3} = 17.56 \times (14.3 - z);$

$p_{CS-4} = 0;$

$p_{CS-5} = 0;$

$p_{CS-6} = 17.56 \times (14.3 - z);$

2. Dry bulk cargo pressure in still water

For load case H: $p_{CW} = \rho_C [0.25 a_X (x - x_G) + K_C a_Z (h_C + h_{DB} - z)]$

$\rho_C = 1$ t/m^3；

$a_X = C_{XG} g \sin\phi + C_{XS} a_{surge} + C_{XP} a_{pitch-x}$

For H1：

$C_{XG}=1;\ \ C_{XS}=-0.8;\ \ C_{XP}=1;$

$$\phi=f_P\frac{960}{L}\sqrt[4]{\frac{V}{C_B}}$$

f_p : Coefficient corresponding to the probability level, taken equal to:

$$\begin{cases}1.0 \text{ for strength assessments corresponding to the probability level of } 10^{-8}\\ 0.5 \text{ for strength assessments corresponding to the probability level of } 10^{-4}\end{cases}$$

$L=180$ m;

$V=14$ kn

$C_B=0.85$

So

$$\phi=1.0\times\frac{960}{180}\sqrt[4]{\frac{14}{0.85}}=10.7442$$

$$a_0=f_P(1.58-0.47C_B)(\frac{2.4}{\sqrt{L}}+\frac{34}{L}-\frac{600}{L^2})=0.4123$$

$$a_{surge}=0.2a_0g=0.2\times0.4123\times9.81=0.81$$

$$\lambda=0.6\left(1+\frac{T_{LC}}{T_S}\right)L=0.6\left(1+\frac{10.8}{12.2}\right)\times180=203.61$$

$$T_P=\sqrt{\frac{2\pi\lambda}{g}}=\sqrt{\frac{2\pi\times203.61}{9.81}}=11.42$$

$$R=z-\min\left(\frac{D}{4}+\frac{T_{LC}}{2},\frac{D}{2}\right)=z-\min\left(9.4,8\right)=z-8$$

$$a_{pitch-x}=10.7442\times\frac{\pi}{180}\left(\frac{2\pi}{11.42}\right)^2(z-8)=0.05676(z-8)$$

$$\begin{aligned}a_X&=1\times9.81\times\sin(10.7442)-0.8\times0.81+1\times0.05676(z-8)\\&=1.1808+0.05676(z-8)\end{aligned}$$

（2）Calculate a_Z

$$a_Z=C_{ZH}a_{heave}+C_{ZR}a_{roll-z}+C_{ZP}a_{pitch-z}$$

$$C_{ZH} = 0.6\frac{T_{LC}}{T}; \;\; C_{ZR} = 0; \;\; C_{ZP} = 1;$$

For H1, $C_{ZH} = 0.6; \;\; C_{ZR} = 0; \;\; C_{ZP} = 1;$

$$a_{heave} = a_0 g = 0.4123 \times 9.81 = 4.0447$$

$$a_{roll-z} = \theta \frac{\pi}{180}\left(\frac{2\pi}{T_R}\right)^2 y$$

$$T_R = \frac{2.3k_r}{\sqrt{GM}} = \frac{2.3 \times 0.35 \times 30}{\sqrt{0.12 \times 30}} = 12.7279$$

$$\theta = \frac{9000(1.25 - 0.025T_R)f_p k_b}{(B+75)\pi}$$
$$= \frac{9000(1.25 - 0.025 \times 12.7279) \times 1 \times 1}{(30+75)\pi} = 25.4230$$

$$a_{roll-z} = 0.1081y$$

$$a_{pitch-z} = \phi \frac{\pi}{180}\left(\frac{2\pi}{T_P}\right)^2 \left|(x - 0.45L)\right|$$
$$= 10.7442 \times \frac{3.14}{180}\left(\frac{2 \times 3.14}{11.42}\right)^2 \left|(x - 0.45 \times 180)\right|$$
$$= 0.05676\left|(x-81)\right|$$

$$a_Z = 0.6 \times 4.0447 + 1 \times 0.05676\left|(x-81)\right| = 2.4268 - 0.05676\left|(x-81)\right|$$

$$x_G = 90$$

$$p_{CW-1} = 1 \times [0.25(1.1808 + 0.05676(z-8))(x-80) + 1 \times (2.4268 - 0.05676\left|(x-81)\right|)(12.5+1.8-z)]$$
$$= 0.25(0.7267 + 0.05676z)(x-80) + (2.4268 - 0.05676\left|(x-81)\right|)(14.3-z)$$

$$p_{CW-2} = 1 \times [0.25(1.1808 + 0.05676(z-8))(x-80) + 1.0918 \times (2.4268 - 0.05676\left|(x-81)\right|)(12.5+1.8-z)]$$
$$= 0.25(0.7267 + 0.05676z)(x-80) + 1.0918(2.4268 - 0.05676\left|(x-81)\right|)(14.3-z)$$

$$p_{CW-3} = 1 \times [0.25(1.1808 + 0.05676(z-8))(x-80) + 1.7897 \times (2.4268 - 0.05676\left|(x-81)\right|)(12.5+1.8-z)]$$
$$= 0.25(0.7267 + 0.05676z)(x-80) + 1.7897(2.4268 - 0.05676\left|(x-81)\right|)(14.3-z)$$

$$p_{CW-4} = p_{CW-5} = 1 \times [0.25(1.1808 + 0.05676(z-8))(x-80)$$
$$= 0.25(0.7267 + 0.05676z)(x-80)$$

$$p_{CW-6} = 1 \times [0.25(1.1808 + 0.05676(z-8))(x-80) + 1.7897 \times (2.4268 - 0.05676\left|(x-81)\right|)(12.5+1.8-z)]$$
$$= 0.25(0.7267 + 0.05676z)(x-80) + 1.7897(2.4268 - 0.05676\left|(x-81)\right|)(14.3-z)$$

3. ***Shear load due to dry bulk cargo***

Still water:

$$p_{CS-S} = \rho_C g \frac{(1-K_C)(h_C + h_{DB} - z)}{\tan\alpha}$$

$$p_{CS-S2} = 1 \times 9.81 \times \frac{(1-1.0918)(12.5+1.8-z)}{0.57735}$$
$$= -1.5598(14.3-z)$$

In Wave

For load case H, R and P: $p_{CW-S} = \rho_C a_Z \frac{(1-K_C)(h_C + h_{DB} - z)}{\tan\alpha}$

For load case F: $p_{CW-S} = 0$

So in this case

$$p_{CW-S2} = 1 \times (2.4268 - 0.05676\left|(x-81)\right|) \times \frac{(1-1.0918)(12.5+1.8-z)}{0.57735}$$
$$= -0.159(2.4268 - 0.05676\left|(x-81)\right|)(14.3-z)$$

$$p_{CW-S} = 0.75\rho_C a_X h_C$$

$$p_{CW-S} = 0.75 \times 1 \times (1.1808 + 0.05676(z-8)) \times 12.5$$
$$= 11.07 + 0.5321(z-8)$$

$$p_{CW-S} = 0.75\rho_C a_Y h_C$$

$$a_Y = C_{YG} g \sin\theta + C_{YS} a_{sway} + C_{YR} a_{roll-y}$$

For case H1 $C_{YG} = 0, C_{YS} = 0, C_{YR} = 0$

$$a_Y = 0$$

$$p_{CW-S} = 0$$

$$p_{BS} = 25$$

4. inner bottom plating

In the longitudinal direction in waves

For load case H1: $p_{CW-S} = 0.75\rho_C a_X h_C$

5. lateral pressure due to liquid

Pressure due to liquid in still water

$$p_{BS} = \max\begin{cases} \rho_L g(z_{TOP} - z + 0.5 d_{AP}) \\ \rho_L g(z_{TOP} - z) + 100 P_{PV} \end{cases}$$

In this case, for fatigue strength assessment,

$p_{BS} = \rho_L g(z_{TOP} - z)$;

In this case, the inertial pressure due to liquid is not considered.

Appendix C

Concepts of optimisation

C.1. Basic concepts in multi-objective optimisation problem

In this research, most of ship design and production process are seen as optimisation process. The design problems are formulated as multi-objective optimisation problems and a new optimisation method has been created when some algorithms have been employed for comparison. In this appendix, the basic concepts in multi-objective optimisation problem are introduced in both descriptive form and mathematical expression for better understanding the optimisation problem.

First of all, the definition of multi-objective optimisation problem is introduced as follows:

Multi-objective Optimisation Problem (MOP): *"a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term "optimize" means finding such a solution which would give the values of all the objective functions acceptable to the decision maker."* (Coello Coello 2007)

In optimisation subject, mathematical expression is more helpful to understand the theory and mechanism of existing algorithms and develop new approaches. In this research, most algorithms provide mathematical model and numerical test functions. So a mathematical definition is provided.

General Multi-objective Optimisation Problem (GMPO): *"A general MOP is defined as minimizing (or maximizing)* $F(x) = (f_1(x),...,f_k(x))$ *subject to* $g_i(x) \le 0,\ i = \{1,...,m\}$ *, and* $h_j(x) = 0,\ j = \{1,..., p\}$, $\mathrm{x} \in \Omega$ *. An MOP solution minimizes (or maximizes) the components of a vector* $F(x)$ *where* x *is a n-dimensional decision variable vector* $x = (x_1,..., x_n)$ *from some universe Ω. It is noted that* $g_i(x) \le 0$ *and* $h_j(x) = 0$ *represent constraints that must be fulfilled while minimizing (or maximizing)* $F(x)$ *and Ω contains all possible* x *that can be used to satisfy an evaluation of* $F(x)$*."* (Coello Coello 2007)

In multi-objective optimisation, finding a direct and unique solution for optimisation is very difficult and the Pareto solutions usually are selected as final solutions. Furthermore, many modern GA algorithms compare the candidate solutions to decide solution space of next step. So, the concepts of Pareto dominance and Pareto optimality are important for optimisation. These concepts were originally in economic research area and have been introduced into optimisation area for seeking a set of solutions to multi-objective optimisation.

Pareto Optimality: "*A solution* $x \in \Omega$ *is said to be Pareto Optimal with respect to* Ω *if and only if there is no* $x' \in \Omega$ *for which* $v = F(x') = (f_1(x'),\ ...\ , f_k(x'))$ *dominates* $u = F(x) = (f_1(x),\ ...\ , f_k(x))$ *. The phrase* ***Pareto Optimal*** *is taken to mean with respect to the entire decision variable space unless otherwise specified."* (Coello Coello 2007)

In Pareto optimality, the notion of 'optimality' is different from single objective optimisation. The good compromise solutions are more reasonable than a single solution.

Pareto Dominance: *"A vector* $u = (u_1, ..., u_k)$ *is said to* **dominate** *another vector* $v = (v_1, ..., v_k)$ *(denoted by* $u \preceq v$ *) if and only if* u *is partially less than* v *,* $\forall i \in \{1,...k\}, ui \le vi \wedge \exists i \in \{1,...k\}, ui < vi;$ *"* (Coello Coello 2007)

Pareto Optimal Set: "*For a given MOP,* $F(x)$*, the* **Pareto Optimal Set***,* P^**, is defined as:* $P^* := \{x \in \Omega \mid \neg \exists x' \in \Omega \; F(x') \preceq F(x)\}$ *"* (Coello Coello 2007)

Pareto Front: *"For a given MOP,* $F(x)$*, and Pareto Optimal Set,* P^**, the* **Pareto Front** PF^* *is defined as:* $PF^* := \{u = F(x) \mid x \in P^*\}$ *"* (Coello Coello 2007)

Weak Pareto Optimality: *"A point* $x^* \in \Omega$ *is a* **weakly Pareto optimal** *if there is no* $x \in \Omega$ *such that* $f_i(x) < f_i(x^*)$*, for* $i = 1,...k$*."* (Coello Coello 2007)

Strict Pareto Optimality: *"A point* $x^* \in \Omega$ *is a* **strictly Pareto optimal** *if there is no* $x \in \Omega$*,* $x \ne x^*$ *such that* $f_i(x) \le f_i(x^*)$*, for* $i = 1,...k$*."* (Coello Coello 2007)

C.2. Basic concepts in Evolutionary Algorithm (EA)

In this research, many concepts and operations are mentioned for proposed system. This part will explain the commonly used concepts in EA.

Individual is a solution to optimisation. In EAs, an individual is often expressed by a string which is coded according to the knowledge of biological genotype in computer environment. One or more chromosomes compose a genotype and a given set of chromosomes is termed a ***population***. In most of EAs, an individual is represented by one chromosome and the set of individuals is a population. The Generation normally is the EA iteration which means the algorithm successively creates a new population. The ***Parent*** in EA means the all the members of current generation when ***Children (or Offspring)*** represents the members in the next generation. The relationships of these concepts are shown in Figure C.1 and Figure C.2.

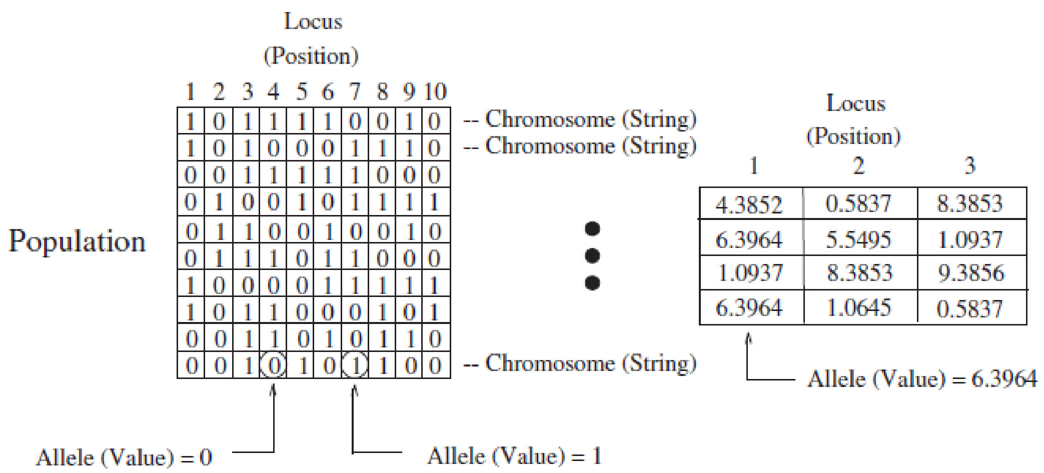


Figure C.1 Generalized EA Data Structure and Terminology
(taken from(Coello Coello 2007))

In EA application, the algorithm often uses two functions: objective and fitness function. These two functions are different in theory. The Objective function is a feature function of the optimisation and the fitness function tell the degree of matching of a solution to the optimisation problem. However, in numerical test functions and practical applications, they are usually, in principle, identical.

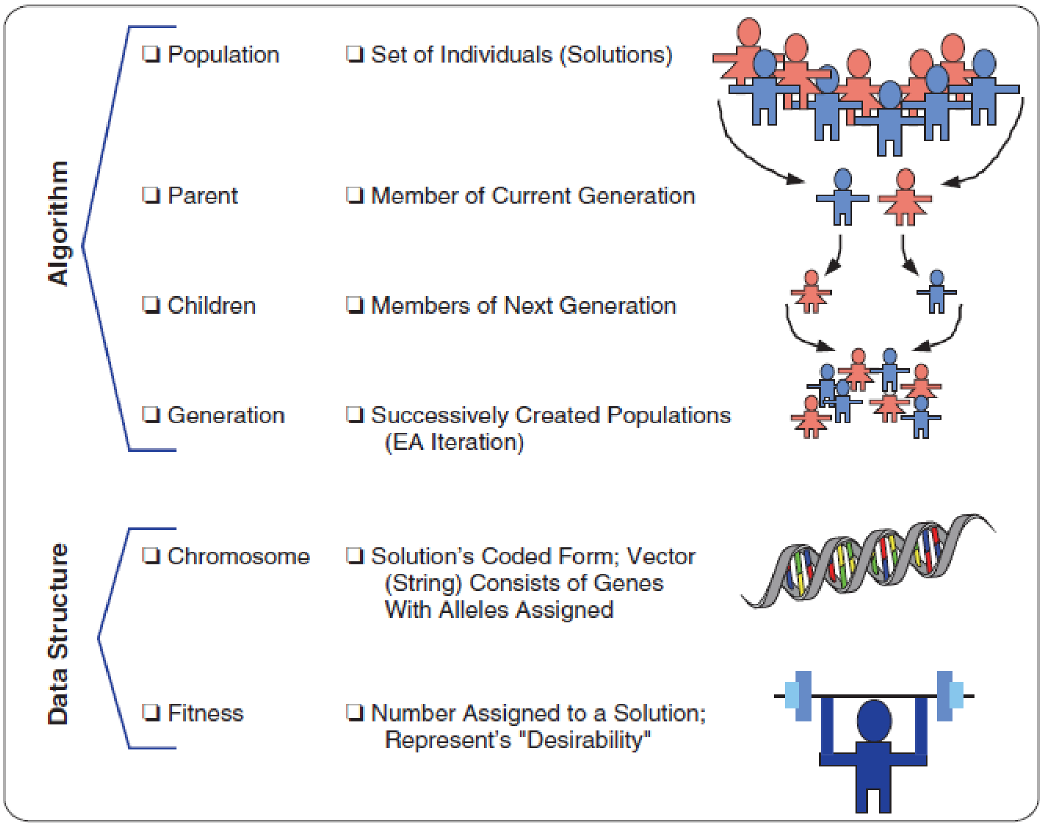


Figure C.2 EA components (taken from (Coello Coello 2007))

Appendix D

Fuzzy multiple attribute decision-making (FMADM) method

D.1. Introduction

The FMADM algorithm deployed in this study is proposed by Olcer in 2005. The whole method can be divided into the following three major states:
1. Rating state,
2. Attribute based aggregation state,
3. Selection state.

In rating state, the expert provides the opinions (or performance ratings) to alternatives according to relative subjective attribute of these alternatives. The ratings given by the expert can be linguistic terms or verbal assessments which are easy to be modelled by fuzzy numbers. Then these data will be converted into standardised positive trapezoidal fuzzy numbers.

In attribute based aggregation state, attribute based aggregation method for homogeneous and heterogeneous (homo/heterogeneous) group of experts is accepted.

In most of situations, the various experts have different importance degree for different practical problem. So heterogeneous (nonhomogeneous) group of experts problem is named for these situations and homogeneous group of experts problem represents that the importance of every expert is same. The degree of importance of experts can be assigned when the weights of attributes are given.

In selection state, all fuzzy elements of the aggregated decision matrices for homo/heterogeneous group of experts are defuzzified. The result of this phase should be a decision matrix with only crisp (or non-fuzzy) data. Then these alternatives will be ranked by selection algorithm.

D.2 Rating state

The aim of this state is to establish the decision matrix for each expert. The estimates of experts of a subjective attribute for an alternative involve subjectiveness, imprecision, and vagueness.

D.2.1 Converting fuzzy data to standardised fuzzy numbers

The linguistic terms in the decision matrix should first be transformed into fuzzy numbers. In this method, the numerical approximation system introduced by Chen and Hwang (Chen 1992) is selected to convert linguistic terms to their corresponding fuzzy numbers. The brief introduction of this system is provided as following:

Experts' fuzzy opinions will be represented as trapezoidal fuzzy numbers. Let U be the universe of discourse where $U = [0, m]$. A quadruplet $A = (a_1, a_2, a_3, a_4)$ can be used to define the trapezoidal fuzzy number. The membership function is

$$\mu_A(x) = \begin{cases} (x - a_1)/(a_2 - a_1) & \text{for } a_1 \le x \le a_2, \\ 1 & \text{for } a_2 \le x \le a_3, \\ (a_4 - x)/(a_4 - a_3) & \text{for } a_3 \le x \le a_4, \\ 0 & otherwise \end{cases}$$

With $a_1 \le a_2 \le a_3 \le a_4$

Assume that each expert $E_k (k = 1,2,...,M)$ constructs a positive trapezoidal fuzzy number $R_k = (a_k, b_k, c_k, d_k)$ to represent the estimated rating for an alternative with respect to a subjective attribute, where $0 \le a_k \le b_k \le c_k \le d_k \le m$.

Translate each trapezoidal fuzzy number $R_k = (a_k, b_k, c_k, d_k)$ into standardised trapezoidal fuzzy number $R^*_k (k = 1,2,...,M)$, where

$$R^*_k = (a_k / m, b_k / m, c_k / m, d_k / m) = (a^*_k, b^*_k, c^*_k, d^*_k)$$

where m is the maximum value of non-standardised trapezoidal fuzzy numbers given by experts for the same attribute.

D.2.2 Attribute based aggregation state

The aim of this state is to create an algorithm to combine a homo/heterogeneous group of experts' opinions to form a group consensus pinion. In this approach, the study proposed in Hsu and Chen (Hsu 1996) is employed.

Assume that the degree of importance of expert $E_k (k = 1,2,...,M)$ is $we_k \in [0,1]$ and $\sum_{K=1}^{M} we_k = 1$.

The aggregation algorithm for homo/heterogeneous group of experts is introduced as follows:

(a) Calculate the degree of agreement (or degree of similarity) $S_{uv}(R_u, R_v)$ is the degree of agreement of the opinions between each pair of experts E_u and E_v. where $S_{uv}(R_u, R_v) \in [0,1]$, $1 \le u \le M$, $1 \le v \le M$ and $u \ne v$;

let A and B be two standardised trapezoidal fuzzy numbers, $A=(a_1,a_2,a_3,a_4)$ and $B=(b_1,b_2,b_3,b_4)$ where $0 \le a_1 \le a_2 \le a_3 \le a_4 \le 1$ and $0 \le b_1 \le b_2 \le b_3 \le b_4 \le 1$. Then the degree of similarity between the standardised trapezoidal fuzzy numbers A and B can be measured by the similarity function S,

$$S(A,B) = 1 - \frac{|a_1-b_1|+|a_2-b_2|+|a_3-b_3|+|a_4-b_4|}{4}$$

where $S(A,B) \in [0,1]$;

(b) Construct the agreement matrix (AM), after all the agreement (or similarity) degrees between experts are measured:

$$AM = \begin{bmatrix} 1 & S_{12} & \ldots & S_{1v} & \ldots & S_{1M} \\ \ldots & \ldots & & \ldots & & \ldots \\ S_{u1} & S_{u2} & \ldots & S_{uv} & \ldots & S_{uM} \\ \ldots & \ldots & & \ldots & & \ldots \\ S_{M1} & S_{M2} & \ldots & S_{Mv} & \ldots & 1 \end{bmatrix}$$

where $S_{uv} = S(R_u, R_v)$, if $u \ne v$ and $S_{uv} = 1$, if $u = v$.

(c) Calculate the average degree of agreement $AA(E_u)$ of expert $E_u (u = 1,2,...,M)$ by using the AM of the problem, where

$$AA(E_u) = \frac{1}{M-1} \sum_{\substack{v=1 \\ v \ne u}}^{M} S(R_u, R_v).$$

(d) Calculate the relative degree of agreement $RA(E_u)$ of expert $E_u (u = 1,2,...,M)$, where

$$RA(E_u) = \frac{AA(E_u)}{\sum_{u=1}^{M} AA(E_u)}.$$

(e) Calculate the consensus degree coefficient $CC(E_u)$ of expert $E_u (u = 1, 2, ..., M)$, where

$$CC(E_u) = \beta \cdot we_u + (1 - \beta) \cdot RA(E_u).$$

where β is a relaxation factor of the proposed method.

(f) Finally, the aggregation result of the fuzzy opinions is R_{AG} as

$$RAG = CC(E_1) \otimes R_1 \oplus CC(E_2) \otimes R_2 \oplus ... \oplus CC(E_M) \otimes R_M,$$

where operators $\otimes$ and $\oplus$ are the fuzzy multiplication operator and the fuzzy addition operator, respectively.

D.2.3. Selection state

After all experts' ratings for each alternative under each subjective attribute, the algorithm needs to rank the alternatives of the problem. All aggregated trapezoidal fuzzy numbers should be defuzzified, so that all components of the aggregated decision matrix are all crisp numbers and any classical MADM method can be used. The selection state consists of two major phases: Defuzzification, and Ranking phases.

(a) Defuzzification phase

In this phase, the fuzzy scoring approach proposed by Chen and Hwang (Chen and Hwang, 1992) is employed to transform all the aggregated fuzzy numbers into numeric ratings. The fuzzy maximising set and minimising set should be first obtained, which are defined as:

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

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

Then, the right score of fuzzy number B can be determined using

$$\mu_R(B) = \sup_x [\mu_B(x) \wedge \mu_{\max}(x)].$$

The left score of B can be determined using

$$\mu_L(B) = \sup_x [\mu_B(x) \wedge \mu_{\min}(x)].$$

Given the left and right scores of B, the total score of B can be computed using

$$\mu_T(B) = [\mu_R(B) + 1 - \mu_L(B)] / 2.$$

(b) Ranking phase

In the ranking phase of the selection state, classical MADM methods can be utilised to determine the ranking order of the alternatives.