SIMULATION OF CONFLICT RISK FOR MARINE TRAFFIC WITHIN A SEAPORT

LI QING

SCHOOL OF CIVIL AND ENVIRONMENTAL ENGINEERING

2013

SIMULATION OF CONFLICT RISK FOR MARINE TRAFFIC WITHIN A SEAPORT

LI QING

SCHOOL OF CIVIL AND ENVIRONMENTAL ENGINEERING

A thesis submitted to Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2013

Abstract

With steady growth of the shipping industry over the past decade, the world’s busiest ports are faced with traffic congestion and potential risk of traffic incidents/accidents in port waters. The causes of traffic incident/accident and traffic congestion come down to a central issue: traffic conflict. Traffic conflicts are prone to occur in port waters due to the special characteristics of port traffic in limited sea space, high traffic density, and complex operational regulations. A conflict is an undesirable event related to safety concerns as well as congestion and delay which affect the efficiency of port operations.

Maritime control centers often play an advisory role, which cannot satisfy the demand on traffic management arising within a busy seaport. Conflict resolution is a critical issue in marine traffic safety, and of great practical significance in traffic congestion management. However, there is no positive control as to systematically resolve conflicts in an effective manner.

This research aims to suggest and provide measures to resolve conflicts for planning and scheduling of port operators so as to minimize delay and congestion ahead of time (for example, in the next day). Two aspects are considered in conflict resolution. Safe navigation of vessels without conflicts is taken as the first priority. The other issue is concerned with reducing the impact of conflicts through minimizing total delay in the traffic network.

Systematic strategy for conflict resolution is a two-stage process. The first stage is referred to as vessel rescheduling for eliminating potential conflicts under original schedules of vessels. An original conflict-free scheduling algorithm is developed in this study for coordinating vessels without conflicts for a given schedule. The second stage is a simulation of vessel movements as a result of the stochastic fluctuations in vessel speed and deviations in timetables in real-time. The applicable corrective measures are identified by a decision-making mechanism. These measures conclude general rules or guidance for navigator chooses proper evasive maneuvers.

Conflict detection as the basis of conflict resolution is the other key objective of this research. Ship domain can be referred to as the clearance area around a vessel, with which the vessel can keep sufficient distance to avoid a conflict with other vessels. An algorithm is presented to predict a potential conflict through an evaluation of positional relation of two vessels’ domains before they actually encounter each other.

A simulation system, named as Marine Traffic Conflict Simulation System, is developed as an experimental platform for evaluating measures for conflict detection and resolution. An application of the simulation system is demonstrated using the seaport of Singapore as an example. Simulation results show that traffic conflicts can be effectively resolved with combination of the scheduling method and the simulation-based corrective measures. Sensitivity analysis is performed for investigation of an influence factor of conflicts: traffic volume. The results show that the number of conflicts and the total delays are sensitive to traffic volume. For instance, within the overall traffic network, as the number of traffic volume changes from -20% to 20%, the number of conflicts changes from -37% to 40% correspondingly. The rate of conflict growth is about twice of the growth of the traffic volume.

The main achievement of this research is a systematic resolution on the practical side for planning for port operators so as to resolve traffic conflicts and congestion ahead of time. Conflict prediction algorithm presents another contribution of this research, as an approach to predict a conflict risk multi-link ahead current vessel position. In terms of practical significance, this research develops an experimental platform (i.e. the simulation system) for evaluating measures for conflict detection and conflict resolution. A study is conducted for Singapore which is the busiest transshipment hub in the world. On the other hand, simulation of marine traffic conflict is a generic study which can be adapted to other busy seaports that are faced with traffic congestion and delays. The logic of conflict detection and resolution is applicable to other traffic systems by changing input data. Therefore, the research is useful for marine traffic safety management in seaports and also helps enhancing the efficiency of port operations.

Acknowledgments

I wish to express my deepest appreciation and gratitude to my supervisors: Professor Henry Fan Shing Leung and Assistant Professor Lam Siu Lee Jasmine, for their excellent guidance, constructive suggestions, and continuous supports during the study of this research topic. Their professional knowledge and skills in analyses of the port traffic network, design of the scheduling method and systematic strategy for conflict resolution have contributed to the success of this research. Their leadership, support, attention to detail, and hard work have also set an example I hope to match in my future life.

I am thankful to Dr Cao Jiaming for his previous work on modeling of the traffic network for the Port of Singapore. His model provides necessary data for reference when I designed my simulation model. I am also grateful to Assistant Professor Wang Zhiwei, with whom I collaborated and had useful discussions on the methods of vessel scheduling.

I would like to thank Associate Professor Wong Yiik Diew, Dr Lum Kit Meng, and all the technicians in Transport & Geospatial Laboratory for the technical supports and for providing a pleasant, productive and academic environment throughout this research.

Special thanks to Associate Professor Liu Yu and Professor Harianto Rahardjo, for their concerns and valuable advices to me when Prof Fan left the school. I would also like to thank Associate Professor Cai Yiyu, with whom I am now working in school of Mechanical & Aerospace Engineering. He has provided me a lot of supports and helps since I had converted to a part-time Ph.D.

Finally, I must thank my parents for their support, understanding and encouragement during my academic years. And in particular, I must acknowledge my husband, Dr Chen Wenyu, without whose love and trust this thesis would not have been finished. It is truly a long journey to complete a Ph.D. I am so lucky that I am not alone in this journey.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

The growth of seaborne trade brings more vessels to a seaport. As traffic volume increases, managing traffic conflicts become crucial. The objective of this research is to evaluate the conflict risk in a seaport and to propose possible solutions. A new simulator is developed dedicated to conflict detection and conflict resolution.

1.1 Background

Maritime transport which carries about 90% of world trade in terms of volume is an important trade facilitator and contributor to economic growth (Kite-Powell 2001). It is the most common transport mode over long distances due to advantages in extensive connectivity and low average cost. Maritime waterways consisting of oceans and seas of the globe cover nearly three-fourths of the world's surface. Goods are transported by waterways to the continents, and then distributed via seaports. Since the 1960s, containerization has improved efficiency in loading/unloading operations, and cargo handling time of maritime transport has been greatly reduced. Although slower than air transport, maritime transport is significantly cheaper, especially for trans-oceanic shipping. Modern maritime transport is a highly effective method for transporting large quantities of goods.

Seaports play an important role in maritime transport. A port is not only the station for vessels to collect supplies and fuel but also the center for cargo distribution. A port acts as an intermediate point providing access to the inland transport system, through which imports and exports transit so as to be traded internationally. In liner shipping, ports also function as transshipment hubs, for handling and temporarily storing cargoes, as well as transferring them to various destinations. Increasing liner shipping facilitates coastal countries to be connected to each other in a global transport network. The good connectivity of transport nodes (i.e. seaports) in the network can contribute to the development of these countries (Lam 2011).

With trade liberalization and increasing globalization, international seaborne trade is boosted by strong growth of world economies in the past decades. Since the 1980s, the annual growth rate of seaborne trade has been 3.7% on average (Grossmann *et al.* 2007). Table 1.1 shows the development of international seaborne trade in the past 30 years. In 2009, the total amount of goods traded showed a minor decline as a result of the global economic downturn and financial crisis. Notwithstanding this, the United Nations Conference on Trade and Development (UNCTAD) in its Review of Maritime Transport (UNCTAD 2011) reported that world seaborne trade in 2010 recovered after the decline and grew by an estimated 7 percent.

Table 1.1 Development of international seaborne trade.

Year	Tanker cargo (Millions of tons)	Dry cargo (Millions of tons)		Total (all goods)
		Total	Main bulks[1]	
1980	1871	1833	796	3704
1990	1755	2253	968	4008
2000	2163	3821	1288	5984
2006	2698	5002	1836	7700
2007	2747	5287	1957	8034
2008	2742	5487	2059	8229
2009	2642	5216	2094	7858
2010	2752	5656	2333	8408

1. Iron ore, gain, coal, bauxite/alumina and phosphate.
Source: Review of Maritime Transport (UNCTAD 2011)

Shipping demand exists when there are needs for shippers to transport their cargoes. Thus, seaborne trade is the driving force of shipping service (Lun *et al.* 2010). Over the past three decades, world seaborne trade grew from 3.7 billion tons in 1980 to 8.4 billion tons in 2010 (Table 1.1). The growth of seaborne trade expresses a clear increased demand to maritime shipping. At the same time, this growth also brings challenges to marine traffic, especially to traffic in port waters. Marine traffic measures the flow of containers in the Twenty-Foot Equivalent unit (TEU) which is a unit of a standard sized container. It is reported that, the number of shipping containers in the global container fleet of equipment has increased from 22.3 million TEUs in 1990 to 32.9 million TEU in 2012. The traffic handling capacity of a given sea space (i.e. a seaport) depends on its infrastructure and operational regulations, as well as on its prevailing traffic pattern (Fan *et al.* 2000). Traffic volume, spurred by frequent vessel movements, may increase beyond the acceptable range, which means the demand for the use of port space may exceed the available capacity. As a consequence, traffic congestion would result.

To gain competitive advantage in shipping service, shippers are focused on reducing transport costs, increasing carrying capacity, and enhancing transportation efficiency. As a result, containerships with larger dimensions are constantly used. It is reported that, the average maximum size of container ships serving all countries between 2004 and 2010 has increased by 66 percent, from 2763 TEUs to 4590 TEUs (UNCTAD 2011). From a seaport perspective, this places intense pressures in terms of port traffic control and navigational safety. The size and the configuration of a large vessel restrict its maneuverability on bendy fairways. As the number of large vessels increases, the possibility of traffic incidents/accidents is likely to increase. According to Rule 7 in the International Regulations for Preventing Collisions at Sea, collision risk exist for a vessel particularly when approaching a very large vessel (Acar *et al.* 2011).

The causes of traffic incidents/accidents and traffic congestion come down to a central issue: traffic conflict. Conflict refers to the situation of near misses between two moving vessels, which occurs frequently in seaports due to the special characteristics of port traffic as follows:

i) Vessels travel along fairways. Fairways are navigable waterways or channels which are open only to vessels with certain draught. Because of the limitations in geographical conditions (width, depth, *etc.*), vessels cannot travel freely in fairways. Conflicts are prone to occur in a narrow fairway, where evasive maneuvers are limited due to insufficient space.

ii) High traffic density. Compared to the open sea, available space within a seaport is limited, while a larger number of vessels move in the traffic network. Port waters often have higher traffic density, especially during the peak period. This poses great potential risk of vessel conflicts.

iii) Complex traffic regulations. Port authorities establish a series of complex regulations for the purpose of controlling and managing traffic. According to geographical conditions, fairways are specified as one-way lane or two-way lane. Vessels are assigned different priorities in operations either to give-way or stand-by. Complex regulations make it difficult for a vessel to take corrective maneuvers in order to avoid conflicts.

A conflict has the most direct effect on safe navigation. Compared with collisions, conflicts do not involve physical contact but relate to the situation of near misses. However, a conflict can also be considered the same as a collision to some extent. The risks resulting from collisions or conflicts only differ in their degree of severity. Conflicts are general incidents, while collisions are dangerous accidents. Collisions present a kind of extreme cases in traffic conflicts. When a conflict cannot be properly resolved, it would lead to a collision accident which could cause a loss of lives and properties, and may threaten the ocean environment.

In general, serious accidents such as collisions are relatively rare. By comparison, the most common consequence of a conflict is time delay which results from the corrective maneuvers of vessels to avoid a collision (Ng *et al.* 2006). The sea space of a busy seaport is finely meshed and intensively used due to increased marine traffic. Within a heavily loaded traffic network, even a small interaction may have a large impact on the entire network. Frequent delays in vessel operations would increase vessel-waiting time and the length of waiting queue, slow down the speed of vessel traffic in the network, and may finally result in traffic congestion.

A conflict is an undesirable event between two vessels related to safety concerns as well as congestion and delay which affects the efficiency of port operations. Vessel conflict is a critical issue in marine traffic safety, and of great practical significance in traffic congestion management. a number of the world's busiest ports in terms of shipping traffic volume are faced with the potential risk of traffic congestion due to increased shipping movement and lack of positive control on vessel movements (Fan and Cao 2000; Grossmann, Otto *et al.* 2007). Maritime control centers often can only play an advisory role in traffic management within port waters. It is therefore desirable to focus research on developing tools that can resolve or mitigate conflicts so as to improve marine traffic conditions within seaports. Based on this background, the study on vessel conflict is proposed in this research.

1.2 Research objectives

This study aims to formulate and provide feasible measures and actions for conflict resolution within a busy seaport. It is emphasized here that this study is to focus on the planning of safe vessel movements within a seaport by the port control center based on known/planned schedules of arriving and departing vessels for the next day as reported and filed by the vessel operators. Therefore, this is not a study for real-time conflict resolution. On the other hand, conflict resolution is not a simple scheduling problem which can be resolved in such a deterministic manner. Vessel movements are stochastic in nature due to random fluctuations in vessel speed, deviation from original schedules. In this sense, random changes in real vessel movements are addressed in a simulation model.

The first important goal of conflict resolution is to enable safe navigation and avoid collisions between two vessels. Given that some delays will be incurred by some vessels in implementing measures/actions to resolve conflicts, attention is also paid to reducing the impact of conflicts by means of minimizing total delay in the traffic network.

For vessel encounters in the sea, taking evasive turns and/or speed adjustment is the most direct way to avoid a conflict. However, the effectiveness of evasive maneuvers depends on whether the risk of a possible conflict could be predicted accurately and timely. To enable effective conflict resolution, simulation system should be able to predict potential conflicts and take corrective measures in advance.

Simulation is an approach to model a real life system on a computer so as to study how the system works. Simulation has good efficiency in integrating complex systems, such as the port traffic system focused in this research; and good performance in computer animation, e.g. to mimic dynamic vessel movements and complex traffic scenarios. This research proposes to develop a simulation system, called “Marine Traffic Conflict Simulation System”, through which key functions of conflict prediction and conflict resolution will be implemented.

The Port of Singapore is the busiest transshipment port in the world and rated as the world's second largest port in terms of container port handling 28.4 million TEUs in 2010 (Yap *et al.* 2013). It is expected that its container throughput will be more than double in the next 15 years. With substantial increases in marine traffic due to the growing shipping industry, the Port of Singapore is faced with traffic congestion and potential risk of traffic incidents/accidents. It has been reported that two collision accidents happened recently in the Port of Singapore in 2009 and 2010, which caused severe damage to human, assets and the environment, especially the collision accident in May 2010 resulted in as serious crude oil slick near the east coast of Singapore. The simulation system will use the Port of Singapore as an example for demonstrating and evaluating the method for conflict detection and resolution.

To achieve these objectives, this thesis has the following tasks:

(1) Design algorithm for conflict prediction.

The ship domain can be referred to as the clearance area around a vessel, with which the vessel can keep sufficient distance to avoid a conflict with other vessels. An algorithm is presented to predict a potential conflict through an evaluation of the relative positions of two vessels' domains before vessels actually encounter each other.

(2) Provide measures/actions for conflict resolution.

The main objective of this research is to provide measures for conflict resolution and to minimize total delay in a traffic network based on a planned schedule of vessel movements. This is achieved by two steps. Firstly, a preplanning process is executed to eliminate potential conflicts under the original schedule of vessel arrivals and departures for a seaport. The second step is to provide proper strategies for resolving conflicts due to random changes in real vessel movements. A simulation model allows one to investigate corrective actions, and whether the planned scheduled needs to be further refined.

(3) Develop a simulation platform.

A simulation platform is developed for the study on conflict prediction and conflict resolution. It provides functions to dynamically display traffic movements, image on impact situation once conflict occurs, as well as other functions for data generation, data processing, simulation control and report output among others.

1.3 Research significance and scope

The main achievement of this research is a systematic resolution on the practical side for planning for port operators so as to resolve traffic conflicts and congestion within a seaport. Conflict resolution is conducted by a two-step process combined with conflict-free scheduling methods in the strategic level and simulation-based corrective measures in the operational level. Development of conflict resolution represents a breakthrough in the field of safe and effective marine traffic control (see limitations and gaps in literature in Section 2.3.3). In regards to traffic congestion management, the resolution strategy focuses on the task of delay minimization so as to improve traffic conditions in a whole traffic network. This is a theoretically interesting and practically useful question.

Conflict detection algorithms are developed as an innovative approach vis-a-vis traditional domain-based method for collision risk estimation. The algorithms solve the domain interference problem by calculation of positional relation of two parallelograms which represent dynamic movements between two vessels’ domains. Conflict detection is the basis of conflict resolution, and also can be served as an individual method for detecting conflicts in any marine traffic environment.

In terms of practical significance, this research develops an experimental platform (i.e. the simulation system) for evaluating measures for conflict detection and conflict resolution. A study is conducted for Singapore which is the busiest transshipment hub in the world. On the other hand, simulation of marine traffic conflict is a generic study which can be adapted to other busy seaports that are faced with traffic congestion and delays. The logic of conflict detection and resolution is applicable to other traffic systems by changing input data. Therefore, the research is useful for marine traffic safety management in seaports and also helps enhancing the efficiency of port operations.

The current simulation system developed in this research is an offline simulation for the traffic control center to execute planning before vessel arriving to the seaport. On the practical side, an online simulation system can be further developed to extend the work into real-time conflict detection and resolution, to point out possible interest of VTS service in port waters.

1.4 Organization of the thesis

This thesis is organized into the following chapters.

Chapter 1 gives a general introduction to the research issues and the objectives of this research.

Chapter 2 reviews the literatures and discusses previous works in several fields related to this research.

In this review, references are made to different approaches proposed in other studies, and the results of these studies are evaluated.

Chapter 3 presents the methodology used in this research. The study consists of three parts: conflict detection, conflict resolution and simulation implementation.

Chapter 4 describes the algorithms formulated to determine and predict a potential conflict between any two vessels during the simulation, which then serves as the basis for conflict resolution.

Chapter 5 describes measures and actions for conflict resolution. A method is developed for preplanning vessels to generate a conflict-free schedule. Strategies or corrective measures are provided in the simulation stage to resolve conflicts resulting from random fluctuations in vessel movements.

Chapter 6 presents the development of the simulation study and introduces the key functional modules in the simulation system. A series of simulation experiments are executed for model demonstration. Sensitivity analysis is also included to evaluate the effect of variations in conflict situations.

Chapter 7 summarizes the findings and conclusions of this research. Possible future works are also proposed.

Chapter 2

Literature review

As mentioned in Chapter 1, conflict detection and conflict resolution are the main objectives of this study. To understand different methods and approaches to the proposed problems, this chapter reviews relevant studies in these two fields. Based on the literature review, the methodology for systematically solving the proposed research problem is presented in the following chapter.

It is noted that almost all past studies are related to marine traffic concern with collision accidents. On the contrary, only a few works addressed the issue of conflict detection and resolution. Conflict in fact can be considered the same as a collision risk to some extent. With regards to navigational safety the risks resulting from collisions or conflicts only differ in their degree of severity. In this view, studies for collision risk detection and collision avoidance will be reviewed as references. Advantages and limitations of the existing studies will be discussed and evaluated.

2.1 Methods for conflict detection

To detect conflicts in a port traffic system, it is necessary to clarify what constitutes a conflict, which criteria could be selected to measure a conflict, and how the measure of conflicts could be quantified. These are attributed to the issue of conflict determination. Only on such a premise, it is feasible to predict conflicts with respect to an established definition of a conflict.

A conflict is expressed by linguistic terms as “traffic interference” (Ng and Wong 2006) or “traffic interaction” (Chauvin *et al.* 2008; Debnath *et al.* 2010). Conflicts are also measured by quantitative criteria. In 1980s researchers designed a radar surveillance sensor for vessel tracking (Isbister *et al.* 1976). In this study, a conflict was defined as an estimated event of intersected paths of two vessels. This criterion was further developed by Ng *et al.* (1998). The researchers identified four conflict situations where paths of two vessels were likely to be intersected. The way of conflict determination was to examine potential occurrence of the identified conflict situations in a certain time period.

Safety separations are also considered as criteria for conflict determination. A study proposed that a conflict was the violation of separation requirement between two vessels (Zhu 2003). The required separation referred to the minimum safety distance along vessel motion. Similarly, researchers developed various separation requirements for conflict determination, e.g. the minimum passing clearances for turning and the minimum bank clearances for anchorage (Ng and Wong 2006).

Existing works in conflict determination cannot be found in literature besides the few discussed above. Importantly, they cannot provide a standard criterion of conflict, because that

- Most of works (Isbister and O'Sullivan 1976; Ng, Lim *et al.* 1998; Zhu 2003) simplify vessels as geometrical points regardless their actual dimensions. This simplification cannot be applied in restricted waters, such as port water where the occurrence of conflicts is sensitive to the size of vessels.
- Some works are particularly developed for a certain traffic environment, e.g. the port of Singapore (Ng, Lim *et al.* 1998) and the terminal of Hong Kong (Ng and Wong 2006). These studies are specific to traffic features of a certain port, and cannot provide a general methodology which can be extended to other cases.

Researchers believe that collisions represent a kind of extreme serious cases in traffic conflicts (Debnath and Chin 2010; Weng *et al.* 2012). A conflict can be considered as a collision risk with a low degree of danger. Studies relevant to collision detection may provide necessary references for conflict detection, and thus are reviewed here.

2.1.1 Collision determination

Criterion for collision determination has evolved from qualitative rules to quantitative mathematical models.

2.1.1.1 International regulations

The International Regulations for Preventing Collisions at Sea (COLREGS) is signed as an international convention which builds the written definition of collision risks (IMCO 1972). Rule 7 in the COLREGS provides a descriptive definition of a risk of collision based on some important considerations in a collision situation (Zec 1996). Three collision encounters are identified in the COLREGS, i.e. head-on, overtaking and crossing (see Figure 2.1), with regard to how vessels approach each other.

Sun (2000) classified collision risks into seven types so as to address some uncommon cases, e.g. a vessel is constrained by its draught and under restricted visibility. Despite of that, the traditional classification is universally accepted. The three kinds of encounters could be transferred to conflict situation, referred as head-on, overtaking and crossing accordingly.

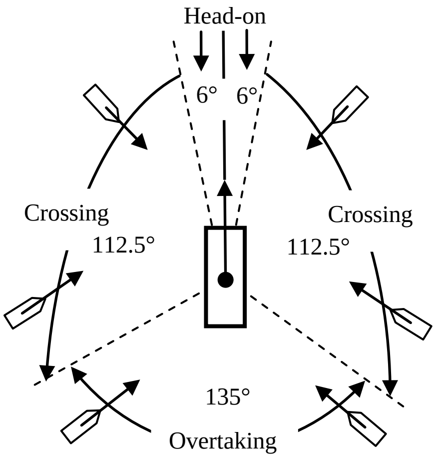


Figure 2.1 Three encounter situations (Zec 1996).

In the COLREGS, a collision risk is expressed according to different degrees of danger. Researchers attempted to explain the COLREGS and thus proposed that there are four stage (see in Figure 2.2) before a collision actually occurs (Cockcroft *et al.* 2003).

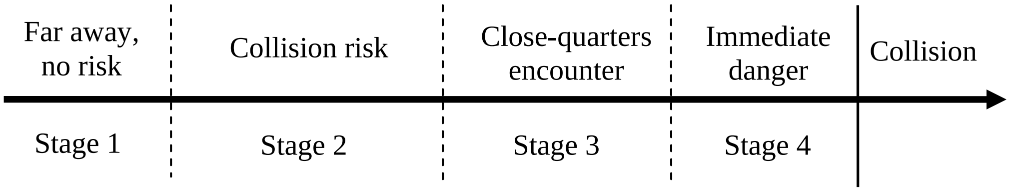


Figure 2.2 Four stages before a collision (Cockcroft and Lameijer 2003).

A study was developed to model the four stages of collision based on geometric analyses of vessel motions (Hu 2001). The boundaries among the four stages are expressed as,

$$DLMA_2 = 20L,$$
$$DLMA_3 = 15L \times \sqrt{1+K^2-2K\times\cos\Delta H},$$
$$DLMA_4 = 5L \times \sqrt{1+K^2-2K\times\cos\Delta H},$$

where

$DLMA_i$: distance between the own vessel and the target vessel when Stage i begins, for i=2, 3, 4,

L: length of the own vessel,

V_0: turning velocity of the own vessel,

K: capability of vessel maneuvering,

ΔH: angle between the own vessel's course and the target vessel's course.

Upon the four stages of a collision risk, it may be inferred that a conflict occurs in the first three stages where the vessels have capability of acting to avoid physical impact. If so, it might be an approach to determine a conflict through distinguishing common collision risks with serious ones. It should be noticed that, however, the four-stage concept is developed with subjective interpretations for the rules in COLREGS. Ambiguity in some rules makes it difficult to precisely build boundaries among different degrees of collision risks. Thus such work is not a sound basis for collision/conflict determination.

2.1.1.2 Closest point of approaches

A quantitative measure of collision risk has been developed using criterion of closest point of approach (CPA). The CPA refers to "the positions at which two dynamically moving objects reach their closest possible distance" (Sunday 2004). Two critical parameters of Distance of closest point of approach (D_{CPA}) and time to closest point of approach (T_{CPA}) are employed. T_{CPA} is often used to estimate the degree of collision risk. For example, a smaller T_{CPA} indicates a higher risk of collision. D_{CPA} is the criterion used to examine whether a collision will occur between two vessels. In a condition of $D_{CPA} = 0$, a collision is certain to happen as long as vessels maintain current courses and speeds.

Monitoring systems such as Automatic Radar Plotting Aid (ARPA) system is capable of tracking vessel positions and acquiring information on the tracked objects' courses and speeds. Upon the information acquired, it is possible to calculate through mathematical modeling for relative movement between two vessels. Researchers developed a model for calculation of D_{CPA} and T_{CPA} with the use of ARPA (Liu *et al.* 2004). A two-vessel encounter scenario is shown in Figure 2.3. The relative bearing of the target vessel is Q, and the distance between the two vessels is d, which can be acquired by ARPA system.

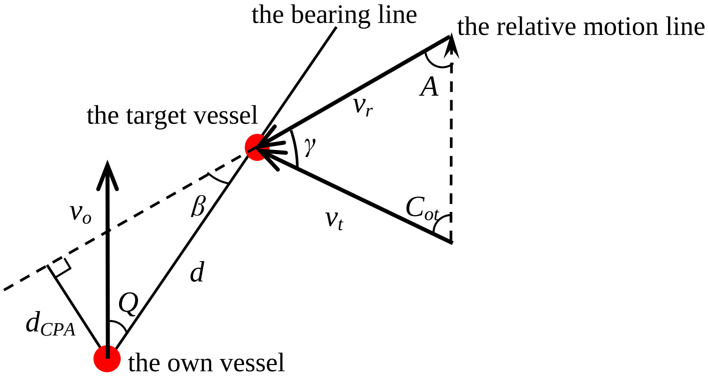


Figure 2.3 Two-vessel encounter scenario (Liu, Wu *et al.* 2004).

Then,

$$\begin{aligned} d_{CPA} &= d \sin \beta, \\ t_{CPA} &= d \cos \beta / v_r \,, \\ v_r &= \left({v_o}^2 + {v_t}^2 - 2 v_o v_t \cos C_{ot} \right)^{1/2}, \\ C_{ot} &= C_o - C_t, \end{aligned}$$

When C_{ot} >= 0°,

$$\begin{aligned} \beta &= A - Q, \\ A &= \arccos\left({v_o}^2 + v_r^2 - v_t^2 \right) \big/ 2 v_o \, v_r \, . \end{aligned}$$

When $C_{ot} < 0^\circ$,

$$\begin{aligned} \beta &= -A - Q, \\ A &= -\arccos\left({v_o}^2 + v_t^2 - v_t^2 \right) \big/ 2 v_o \, v_r \, , \end{aligned}$$

where

β: angle between the relative motion line and the bearing line,

A: angle between the relative motion line and the heading line of the own vessel,

V_t, v_o: speed of the target vessel and that of the own vessel, respectively,
C_t, C_o: course of the target vessel and that of the own vessel, respectively,
v_r: relative speed between v_t and v_o.

Parametric analyses of CPAs are commonly used to estimate collision risk and assist with decision-making in collision avoidance. CPAs represents an easy-to-implement criterion of collision determination. Calculation of CPAs is based on an assumption that vessels are simplified to geometrical points. This assumption is reasonable for radar tracking in open sea environment, considering two approaching vessels are far away so that their actual sizes are ignored. However, it has an inherent limitation for restricted waters.

2.1.1.3 Ship domain

The CPA criterion is difficult to use in restricted waters such as narrow fairways where a vessel cannot choose routes freely and its maneuvering is restricted. In view of this, the concept of ship domain is proposed as “a water area around a vessel which is needed to ensure the safety of navigation and to avoid collision” (Zhao *et al.* 1993). Ship domain was first presented by Fujii, *et al*. (1971). Their study established a ship domain model for a narrow channel based on field observations. Later, Goodwin (1975) developed a ship domain model for open sea and analyzed how traffic density and length of vessel affect the size of the ship domain.

Since then, many studies focused on improving models of ship domain (Davis *et al.* 1980; Coldwell 1983; Zhu *et al.* 2001). At present, there are three main methods for modeling ship domain: statistical method, analytical method and artificial intelligence method. Statistical method is developed to model a ship domain based on statistics data of vessel positions and motion trajectories. Analytical method aims to describe the boundary of ship domain as a function of variables which represent important factors in navigational safety, such as dimensions, course, speed or relative speed of vessels.

Artificial intelligence method is applied in modeling of fuzzy ship domain. The concept of fuzzy ship domain is based on the idea that the boundary of ship domain is not fixed and is affected by many factors such as sea area, traffic density, the maximum speed and length of a vessel. (Tam *et al.* 2009). In previous studies, the definition of ship domain theoretically classified the area around a vessel into two zones: safe and dangerous. Domain of a vessel is referred to as the safe zone, in contrast with the dangerous zone outside it. In practice, however, navigators are required to distinguish a larger number of zones with different degrees of danger, so as to select one in which an acceptable level of safety is maintained. Hence researchers proposed the fuzzy domain boundary using the theory of fuzzy sets

(Zhao, Wu *et al.* 1993; Pietrzykowski 1999; Pietrzykowski 2008; Wang 2010). Pietrzykowski (2008) developed a series of fuzzy domains to model levels of navigational safety based on navigators' knowledge. Domains of a vessel along a straight fairway are described in Figure 2.4, where parameter *γ* represents navigational safety level.

Modeling of ship domain is based on the considerations of vessel speed, dimensions, maneuvering characteristics and even traffic conditions. Compared with the CPAs, it is a more comprehensive and accurate criterion for determination of collision risks as well as conflicts in port waters. On the other hand, ship domain is a relatively complicated parameter that may not easy to be modeled, especially concerning its fuzzy form.

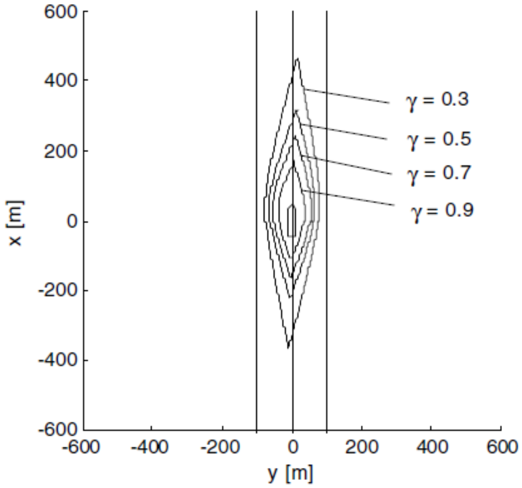


Figure 2.4 Domains of a vessel along a straight fairway (2008).

2.1.2 Prediction for collisions and conflicts

Risk assessment is a useful tool for risk analysis, while also has significance in risk prediction to some extent. Risk assessment methods are developed for estimation of risks of collisions and conflicts. It is noticed that, risk assessment is not a prediction but represent the offline estimation for the prevailing trend of risk, i.e. possible probability of collisions and conflicts, based on historical data and statistical analysis.

Risk forecast is an approach to monitoring and forecasting real-time risks of collisions and conflicts. Risk forecast studies have been developed with critical technologies of radar surveillance or tracking, satellite-positioning, radio communication, and data handling. As a critical issue of traffic control, risk forecast for marine traffic accidents, such as collisions and conflicts, has been widely implemented by both independent bodies (i.e. vessels) and harbor/port authorities.

2.1.2.1 Risk assessment

Risk management is a process of identify, estimate and solve (by means of mitigation measure, risk management plan or tool) the undesirable event likely to occur. Under risk management, risk assessment is mainly employed for two purposes: estimating probability of occurrence of the event, and evaluating its causes and consequences. Among them, risk estimation represents an approach to predicting risks of collisions and conflicts.

As early as in the 1970's, risk assessment method was applied for evaluating marine accidents (i.e. collision and grounding) (Macduff 1974). From then a number of methods have been developed for collision risk estimation, such as statistical method, fault tree method, data analysis, mathematical model, system simulation and expert judgment (Judson 1992; Bruzzone *et al.* 2000; Friis-Hansen *et al.* 2002; Merrick *et al.* 2002; Pietrzykowski *et al.* 2002; Merrick *et al.* 2003; Wang *et al.* 2004; McGeoch 2005; Morel *et al.* 2006; Szlapczynski 2006; Hu *et al.* 2007; Trucco *et al.* 2008). These approaches are categorized into two types: descriptive analysis and statistical modeling.

Descriptive analysis

Fault tree analysis (FTA) and event tree analysis (ETA) are typical descriptive analyses by using a logic graphical process to evaluate risks. FTA and ETA studies are developed for investigating potential collision accidents and estimating likelihood of them (Ronza *et al.* 2003; Trucco, Cagno *et al.* 2008; Yao *et al.* 2010; Kader *et al.* 2011).

An example of an event tree is shown in Figure 2.5. The event tree is constructed by an initial event and its possible consequences using a forwards logic. The probability of the critical event (i.e. near misses) is estimated on failure rates of other events in the tree. The idea of FTA is similar to ETA, but follows a backwards logic to quantify events. ETA and FTA are semi-quantitative risk estimation method due to subjective nature in processes of hazard generation in the tree.

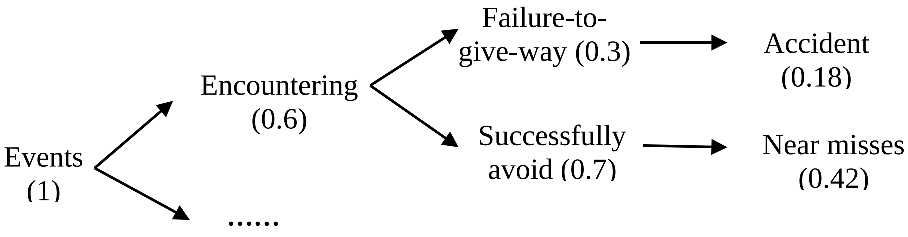


Figure 2.5 An event tree (Ronza, Félez *et al.* 2003).

Statistical modeling

Statistical modeling is developed for formalizing accident probability based on certain statistical assumptions of random variables. Several studies on collision and conflict risk modeling are discussed below.

Hashimoto and Okushima (1990) built models to estimate collision risks in channels under different encounter situations (i.e. head-on, overtaking, overtaken and crossing). Take head-on situation as example, referring to an encounter of two vessels from the opposite directions. The probability of collision was calculated as the product of conditional probabilities of the give-way failure probability and the confrontation probability. The basic model is described as follow,

$$P[C|E]^h(i,j;t) = P[F|E]^h(i,j;t) \bullet P[C|F]^h(i,j),$$

where

$P[C|E]^h(i, j; t)$: the probability that vessel i collides with a vessel j in time period t,

$P[F|E]^h(i, j; t)$: the probability of confrontation,

$P[C|F]^h(i, j)$: the probability of give-way failure.

In the above model, the tracked locations of vessels were supposed to follow normal distribution. The probability of give-way failure could be determined through analysis of the relative motion between vessel i and vessel j. The probability of confrontation could be calculated according to the distribution of relative distance between them.

A study for conflict risk assessment was developed by Zhu (2003). The conflict risk model was built using similar methodology adopted by Hashimoto and Okushima (1990). For example, the probabilistic conflict risk in a two-way channel was estimated through calculation of the probability of non-conflict with given distributions in speed and arrival pattern of vessels. The model is expressed as

$$P_{cij} = 1 - F_{nij},$$

$$F_{nij} = \frac{1}{t_{\max}} \int_0^{T_{\max}} f_{nij}(t)\,dt,$$

$$f_{nij} = \left(1-e^{-\lambda t}\right)\cdot\left[F_D\left(D > D_{ij}\right) + F_0\left(0 < T < \infty \middle| D < D_{ij}\right)\right] + (e^{-\lambda t}),$$

where

P_{cij}: the probability that vessel i collides with a vessel j,

F_{cij}: the probability of non-conflict between vessel i and vessel j,

f_{cij}: the probability density function for no conflict,

T_{max}: a reasonable maximum value of the time period required to traverse a channel,

F_D ($D>D_{ij}$): the probability of the distance between two successive vessels is smaller than the minimum separation required,

F_0 ($0<T<\infty|D<D_{ij}$): the probability of vessel j being able to overtake.

Collisions as serious accidents are rarely observed, especially in a certain region, e.g. the water within a seaport. It is difficult to obtain sound statistic data from existing collision accidents so as to construct statistical models for risk estimation. To overcome the low sample problem, researchers developed an indirect approach to collision risk estimation by measuring a serious conflict as an indicator of collision risk (Debnath and Chin 2010). Quantitative measurement of a conflict is a reasonable alternative to collision risk estimation due to the advantage of sufficient data. The basic model of collision risk estimation is expressed as

$$P_c = p(C'_{\max} > \tau) = 1 - F_\tau(\tau),$$

where

P_c: collision risk of a waterway,

C_{max}': the severity of conflict,

$F_\tau(\tau)$: the cumulative distribution function of non-serious conflicts

τ: a threshold value of C_{max}' that distinguish serious conflicts from non-serious ones.

In the above model, the threshold value was calculated by a five-point conflict risk scale which was established by expert judgments from a survey on harbor pilots. Upon this risk scale, conflicts with the highest risk were perceived as serious conflicts, i.e. collision risks. However, this way to determine collision risks is a subjective estimation and its accuracy greatly relies on experts' experiences and judgment. It is desirable to use a more objective criterion for distinguishing serious collision risks from conflicts.

There are other models developed for collision risk assessment, such as a physics-based Markov model (Tan *et al.* 1999), an ordered probit regression model (Chin *et al.* 2009) and a geometrical model with optimization algorithms (Montewka *et al.* 2010). An overview of risk assessment models for maritime waterways has been presented by Li, *et al.* (2012).

2.1.2.2 Risk forecast

Risk forecast represents an online prediction technology involving issues of traffic surveillance, vessel tracking, vessel positioning, information communication and data handling. These parts could work separately or combined with each other for the purpose of predicting incidents/accidents. A overview of critical equipment and technologies for risk forecast is given in the following (Williams 1988):

- Radar: Marine radar plays an important role in navigation and positioning. Equipped with radar, a vessel can receive navigation data on other vessels. The deployment of harbor radar is a vital element of Vessel Traffic Services (VTS). It provides the port authorities the necessary information to offer effective services. It also enables constant monitoring of the movements of vessels within the port. The effectiveness of radar is limited by environmental conditions. When the traffic density is great and/or weather is bad, radar data may be lost.
- Very High Frequency (VHF) radio: VHF radio has been employed in the past for communication of marine traffic from a shore-based station or stations. When unacceptable vessel movements occur or are about to occur, VHF radio will alert vessels. The application of VHF has been endorsed by the International Maritime Organization (IMO).
- Automatic Identification System (AIS): AIS is a ship-borne automatic broadcasting system, whose essential technology is Self-Organizing Time Division Multiple Access (SOTDMA). It is applied in the communication between vessel and vessel, as well as vessel and shore. The use of AIS for radar surveillance generally, and for VTS in particular, has been recognized for a long time. The original IMO timetable allowed a considerable period for the fitting of AIS equipment. By the end of 2004, virtually all commercial vessels have been fitted with AIS (Stitt 2004).
- Electronic Chart Display and Information System (ECDIS): As a new kind of vessel navigation system and decision-making system, ECDIS could continuously monitor vessels, provide information related to navigation, and predict risks to avoid incidents/accidents. At present, more than 200,000 vessels are equipped with ECDIS. It is an inevitable trend that paper nautical charts will be replaced by ECDIS. Combined with Geographical Information System (GIS) and Global Positioning System (GPS), a dynamic surveillance system for collision predication has been developed (Zhu *et al.* 2004).

Plenty of studies focus on application and development of technologies in collision risk forecast (Isbister and O'Sullivan 1976; Degre 1995; Hughes 1998; Cockcroft 2003; Stitt 2003; Zhu and Liu 2004; Cairns 2005; Kingsley 2006; Li *et al.* 2007). Among them, an important contribution is VTS-based collision alerting systems which are established by port authorities for monitoring, controlling port traffic and enabling navigation safety. Recently the focus of VTS is on development of VTS/AIS/GIS integrated traffic control systems (Harre 2000; Kao *et al.* 2007). Such integrated systems provide services for precise collision prediction, such as forecasting of collision time and position, so as to aid navigator’s decision-making of collision avoidance.

2.2 Methods for conflict resolution

To the best of this researcher’s knowledge, no systematic work has been developed for resolving marine traffic conflicts. Several studies in marine traffic management are concerned with conflicts, but only pay attention to resolve them by risk forecasting technologies such as traffic monitoring aid systems and other vessel service systems (Isbister and O'Sullivan 1976; Degre 1995).

Collision avoidance is a method for helping navigators to choose proper maneuvers when faced with a collision risk and enhance general intuition on how to operate in similar situations. Studies of collision avoidance would provide evasive strategies for resolving real-time collisions/conflicts. On the other hand, collision avoidance focuses safety of the own vessel in direct control of navigator, but does not pay attentions to the behavior of whole network traffic. Planning methods developed in marine traffic control are concerned with safe and congestion free traffic within the whole network. Studies relevant to conflict-free planning can be considered as an approach to conflict resolution and thus are brought up for discussion here.

2.2.1 Collision avoidance

The COLREGS is the first international agreement established for mariners in collision encounters and as a legal basis when dealing with collision accidents. It represents the international convention that has the most immediate significance in instructional guidance for collision avoidance.

The COLREGS attempts to provide a series of instructions to cope with every conceivable encounter, but in some cases the rules are unclear or ambiguous. Since the COLREGS is published, the topic of removal of ambiguity in the rules has been discussing (Clements 1998; Millns 1998; Weber 1998; Kemp 2002; Manley 2002; Salinas 2002). Therefore, when the rules are applied in practice, mariners need “an experienced eye for how the individual components should be interpreted in choosing what to do” (Taylor 1998).

The COLREGS specifies technical standards in human-vessel navigation, but it is hardly applicable for determining precise avoidance maneuvers in complex collision scenarios. Alternatively quantitative mathematical methods are developed. In previous studies, the focus of collision avoidance can be categorized into two fields:

- Evasive maneuver,
- Evasive trajectory.

2.2.1.1 Evasive maneuver

For an individual vessel (i.e. own vessel), taking maneuvers of course alteration and/or speed adjustment is the direct way to avoid colliding with others. The evasive maneuvers are usually determined according to required CPA indicators. Take a two-vessel encounter scenario as example (Figure 2.6). It is assumed that the own vessel (OV) has a collision risk with the target vessel (TV) under current DCPA. To avoid the collision, the OV needs to alter course so as to enable safe distance of closest point of approach (SDCPA) with the TV.

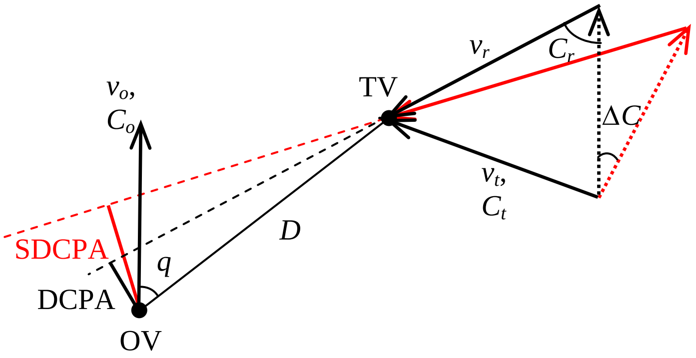


Figure 2.6 Course alternation for collision avoidance (Li *et al.* 2001).

The magnitude of course alternation (ΔC) can be calculated by geometrical analysis of relative motion between the vessels. A mathematical model (Li and Hu 2001) is described as follow,

$$\Delta C = \left[\sin^{-1}\left(SDCPA/D \right) \pm \left(C_r - q \right) \right] \cdot f\left(k, \Delta H \right),$$

$$f\left(k, \Delta H \right) = \frac{1 - 2k\cos\Delta H + k^2}{1 - k\cos\Delta H},$$

$$k = v_t / v_o,$$

$$\Delta H = C_o - C_t,$$

where

D: the distance between TV and OV,

q: the relative bearing between TV and OV,

C_o, C_t: the course of OV and TV, respectively,

v_o, v_t: the speed of OV and TV, respectively,

C_r, v_r: the relative course and the relative speed between TV and OV, respectively.

Similar studies are developed for collision avoidance with CPA-based indicators (Lenart 1983; Zhao *et al.* 1991; Lenart 1999; 2000; Liu, Wu *et al.* 2004). The idea of such studies is to determine the evasive maneuver with a mathematical model that constructed with speed and course of the two closing vessels concerned. Application of different models depends on accessible parameters in actual collision situation.

Lenart (1983) theoretically proposed a cone-shaped collision danger region outside which the own vessel could avoid colliding with targets. The collision danger sector (CDS) represents an extended CPA-based indicators for evasive maneuvers, which was qualified by Pedersen *et al.* (2003). The researchers built a CDS model with a parameter of the minimum passing distance, i.e. CPA_{lim}. As shown in Figure 2.7, the CDS is referred to as the shadowed region consisting of the bold dash lines. The own vessel with initial velocity (v_o) will collide with a target vessel with velocity (v_t) since the distance at CPA is zero. Any maneuver that makes velocity vector of the own vessel outside CDS is bound to effective evasive maneuver. For example, when an evasive maneuver $v_{o\text{-}1}$ alters to the fringe of SDS, the two vessels will safely pass through each other with the CPA_{lim}.

With regards to the concept of ship domain, a collision risk can be indicated by the event of domain violation between two vessels. Szlapczynski (2006) developed the domain-based indicator for determining an evasive maneuver, which was consisting of two parameters: the temporary approach factor of own vessel ($f(t)$) and the approach factor for a given situation of approach to the target vessel (f_{min}). The study proposed a numerical algorithm to calculate the approach factors corresponding to various ship domains. Simulation results proved that the domain-based approach factors, compared with DCPA parameter, encouraged more precise judge of collision risk while enabling evasive maneuvers conformed to the COLREGS.

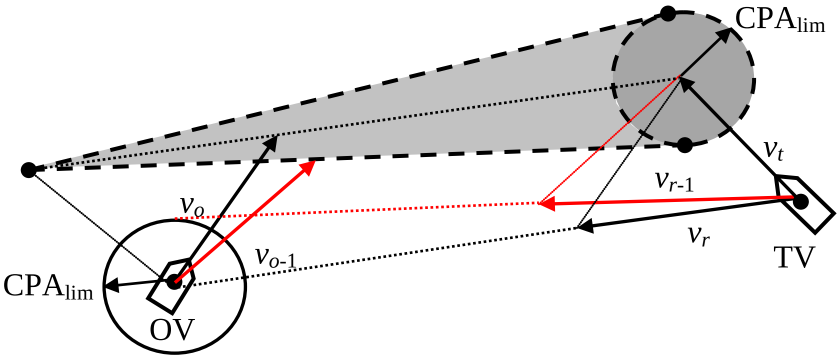


Figure 2.7 Collision danger sector (Szlapczynski 2006).

Decision-making models

In case of a collision risk, navigator has many choices of evasive maneuvers which are theoretically possible to avoid the collision. However, not all candidates are acceptable in practice. For example, the range of course alternation should not be less than 15 degrees in accordance with Rule 8 in COLREGS, and not be larger than 60 degrees due to economic considerations (Szlapczynski 2006). Decision-making model is hereby developed for the selection of a proper and/or optimal evasive maneuver.

Methodologies in collision avoidance decision-making can generally be grouped into classical mathematical methods (e.g. mathematical model) and soft-computing techniques (e.g. Artificial Intelligence (AI)). The mathematical method involves precise mathematical description of vessel maneuvers, vessel dynamics and vector of motion (Browning 1991). Mathematical models normally are developed based on a sequence of strict definitions or assumptions. For example, Zhao and Wang (1989) proposed a model for deciding the starting time to take an avoidance action. Another research (Wu *et al.* 2000) established an action time model with analyses of the last opportunity point to steer for a vessel in a collision situation.

The area of AI is related to expert systems, evolutionary algorithms, fuzzy logic, and neural networks (NN), as well as combinations of them. The first application of expert system was a decision-making system for collision avoidance (Hayama *et al.* 1987). The use of expert system alone may not guarantee timely and precise decisions. A proper decision-making process requires a combination of additional methods. For example, neural network has good efficiency in learning capabilities (Inaishi *et al.* 1992; Liu *et al.* 2005); fuzzy logic can simplify complex computations due to its high mathematical abstraction (Lee *et al.* 2004); evolutionary algorithms solve collision avoidance problems by exploiting their optimization capabilities (Smierzchalski *et al.* 2000). The use of one or a combination of these methods in accordance with requirements may help the navigator in charge to make proper decisions (Wu and Zheng 2000).

Traditional decision-making models are developed on the idea that selecting an optimal option with the maximization of expected attributes from available choices. However in practice, navigator may not make many options due to “time pressure, ambiguous information, ill-defined goals and changing conditions” (Zsambok *et al.* 1997). Recognition-primed decision (RPD) strategy was developed for naturalistic collision avoidance decision-making (Chauvin and Lardjane 2008). The basic frame of RPD was consisting of three functions: simple match, evaluate the evasive maneuver, and diagnosis of a situation. These functions constitute a quick and proper decision-making process for navigator. From a practical viewpoint, RPD is a useful tool for aiding navigator to improve cognitive capacity in potential collision situations.

2.2.1.2 Evasive trajectory

Human error in vessel-handing is a critical cause of collision accidents. Since the 1980s, researchers have focused on development of autonomous navigation system as a promising approach to reduce human errors and thereby avoid collisions. An autonomous vessel is capable of selecting and planning of an evasive trajectory to automatically avoid a collision with targets. Evasive trajectory represents a series of evasive maneuvers generated in a varying collision situation.

The approaches to the evasive trajectory problem are various. Lisowski and Smierzchalski (1995) used a finite-dimension nonlinear programming method to determine the evasive trajectory with the use of a maximum principle. Furuhashi, *et al.* (1996) developed a genetic algorithm to estimate the evasive trajectory through modeling a fuzzy process for the collision situation.

The above studies were developed based on an assumption that the target vessel remains its speed and course during the process of collision encounter. To overcome this limitation, Smierzchalski and Michalewicz (1998) took into account variable speed of target vessels for the evasive trajectory problem using an evolutionary method. In this study, the task of searching a safe trajectory was preformed with respect to fixed navigational constraints and dynamic constraints such as time parameter, variable speed of the moving vessels. Comparison tests proved that the evolutionary method was flexible to control online planning with a reasonable computational time.

Besides safety concerns, evasive trajectory routing is also be regarded as an optimal task. Chang *et al.* (2003) proposed a maze algorithm for searching a most economical evasive trajectory on raster charts. The principle of maze routing was to search the shortest path between a given pair of cells on a rectangular grid of cells without crossing any obstacles. This routing algorithm was capable of performing various speeds for multiple vessels and finding optimal routes for varied terrain in the raster chart.

The use of the shortest path algorithm still has limitations for its results are not always the same as optimal. Szlapczynski (2005) proposed an improved strategy to search an optimal shortest path with penalty for bends, which can be treated as minimizing the number of turns within a route. The author designed a new data structure to reflect upon the cost of all course alterations in each cell’s arrival time. Meanwhile speed maneuvering was considered as an alternative to the course maneuvering for collision avoidance. A search algorithm was used to determine the minimum reduction of speed required to avoid the collision.

A review of methods for evasive trajectory was developed by Tam *et al.* (2009). The authors pointed out that the evasive trajectory problem had been theoretically solved, but remained many limitations in efficiency and completeness for practical implementation.

2.2.2 Conflict-free planning

Port water is characterized by a traffic network where vessels move in fairways with given routes and schedules. The shore-based system with VTS provides services of vessel tracking and information transmission between traffic controller and vessels. In traffic control point of view, it is possible to develop and apply planning methods for resolving conflicts within a seaport.

Conflict-free planning belongs to a traffic control problem, which have been developed in context of rail and air traffic control (Slattery *et al.* 1994; Tomlin *et al.* 1998; Hatzack *et al.* 2001; Pallottino *et al.* 2002; Eyferth *et al.* 2003; Doniec *et al.* 2008). However, studies for conflict-free planning cannot be found in the maritime field. Alternatively, some vessel routing and scheduling methods that contribute to marine traffic control are reviewed as references.

2.2.2.1 Routing methods

Marine traffic control can be treated as a task of vessel route assignment. It is believed that the encounter rate of vessels reflects the probability of collision (Anderson *et al.* 1996). The assignment of route aims at allocating a most suitable route for each vessel so as to reduce overall encounters of vessels passing through a given area. A study for vessel routing problem was developed to resolve collision risks in channels and waterways (Filipowicz 2004). Certain areas prone to generate collision risks were identified with the maximum acceptable capacities. Any route of a vessel was associated with a cost value. Basic idea of the route assignment was based on minimizing the overall cost value of vessels while maintaining allowable capacities for the areas concerned. This is the NP-complete class of generalized assignment problem. A learning evolutionary algorithm was applied to solve this problem.

In Filipowicz (2004)'s study, the number of encounters involved in a route reflected the risk factor of a collision. However this criterion is insufficient for the case that the number of encounters is the same for two and more alternative routes. For this concern, researchers designed a vessel routing system with multi-criteria for risk estimation, such as hydro-navigational conditions, number of encounters in different collision situations (Szlapczynska 2005; 2006). In the routing system the best route for a vessel was selected from a group of available routes according to a route ranking generated by a fuzzy method.

The routing methods discussed above are developed to support centralized traffic control by VTS systems. Under the centralized control strategy, vessels within a water area communicate with the control center to get information and guidance for safe navigation. Nevertheless the traffic control center cannot acquire complete information of all elements (e.g. vessels) in the traffic network. For efficiently communication and cooperation among vessels, a distributed problem solving architecture, i.e. multi-agent traffic control, is proposed. This distribution of traffic control treats vessels as a group of intelligent agents connected with each other via a negotiation mechanism. Thus, vessels could communicate with each other and each vessel could make her decisions for anti-collision actions. The expectation-based negotiation protocol is suitable for vessel routing in the cases with complicated scenarios and uncertainties.

Jin *et al* (1990) designed and implemented a vessel routing system with a distributed traffic control. Each agent (i.e. a vessel) in the traffic system was responsible to select a safe route without collision risk with others. The route was made with local and global planning according to safety concerns for itself as well as interactions with other vessels. The basic routing process can be described as: detect collision risks firstly; if an encounter occurs, decide collision avoidance maneuvers; negotiate with others and make iterative co-routine for safe one.

The use of routing methods is based on an assumption that a vessel can free to choose its route from a group of available ones. However for vessels in port water with specified routes, their choices for alternative paths are limited. Thus, the routing methods discussed above would be difficult to be applied in practice. It is also noticed that, the routing problem is indeed a complex computation process. The above methods are tested with a few vessels in simple scenarios, which cannot prove its robustness for a huge number of vessels. The practical implementation of the routing methods requires a good balance between efficiency and available computer power.

2.2.2.2 Scheduling methods

Vessel scheduling is the assigning of times or time intervals to various events of vessels. There are two kinds of studies developed in vessel scheduling. Shipping scheduling is developed for solving shipping problems with optimal tasks, e.g. maximize profits per time unit and minimize fuel costs for liner and industrial transportation. Studies on shipping scheduling are interested in improving economies and efficiency of shipping operations. Reviews of shipping scheduling models and methods were presented by David (1983; 1993).

Traffic scheduling is another topic addressed for marine traffic management. Researchers developed vessel transit scheduling algorithms for aiding VTS on sequencing vessel entrances in channels and straits (Ulusçu *et al.* 2009; Ulusçu *et al.* 2011). The scheduling of vessel transit is supposed to arrange passage traffic while enabling the safety of navigation. In Uluscu (2009)'s study, a mathematical model was developed for risk analysis for current scheduling practice in strait of Istanbul. The principle of the scheduling was to schedule the vessels with highest waiting time first while enable priority to large vessels carrying dangerous cargo. Application of the scheduling algorithm can also be extended to other straits and channels.

Other researchers focused on development of intelligent agent systems for scheduling of vessel berth and sailing in a container terminal (Lokuge *et al.* 2007). This scheduling system was designed to manage terminal traffic so as to improve the productivity and efficiency of berths. The neural network-based BDI agent architecture supported various fuzzy plans for agents in complex dynamic scenarios in a container terminal.

Studies for marine traffic scheduling cannot be found in literature, besides the few works discussed above. The existing scheduling methods are not developed for conflict free traffic planning. However, dealing with issue of traffic conflict, these routing methods can also be executed with a corresponding method for conflict detection and avoidance by only changing or adjusting constraints in the scheduling task.

2.3 Summary

The review of past studies is consisting of two parts: conflict detection and conflict resolution.

2.3.1 Conflict detection

A clarified definition is the criteria of conflict detection. In the COLREGS a collision risk is classified into three encounter situations. This classification can be transferred directly to conflict situations. The COLREGS also describes a collision risk as a process of different degrees of danger. A collision risk with low degrees of danger may be referred to a conflict. Nonetheless it is rather difficult to distinguish degrees of a collision risk.

Parametric analysis of CPAs is widely used to determine collision risks in open sea, but it is restricted in port waters. Alternatively, ship domain is proposed as a more comprehensive and accurate criterion for determination of collision risk with consideration of vessel dimensions. In regards of a conflict situation, a ship domain represents a water area around a vessel which enables her to keep necessary clearances with targets. The shape and size of the ship domain are affected by factors such as vessel speed, dimensions, location in the sea space network, and traffic density.

Risk assessment and risk forecasting are two main directions in studies of collision and conflict prediction. Assessment methods are used to estimate the probability of collisions/conflicts in a given traffic network. Use of the assessment method relies on available historical and statistical data. Risk forecasting aims to detect and monitor real-time risk for vessels. The technologies in risk forecasting provide necessary conditions for shore-based port traffic control with VTS.

2.3.2 Conflict resolution

Two categories of studies are reviewed as references for conflict resolution: collision avoidance and conflict-free planning.

Collision avoidance provides guidance to own vessel with proper evasive maneuvers in a collision encounter. Regulations for collision avoidance (e.g. the CLORGES) represent technical standards in human-vessel navigation. To determine precise evasive maneuvers, quantitative criteria, e.g. CPA and ship domain, are applied. Researchers develop decision-making models for choosing a proper or the optimal option from all available evasive maneuvers. Single evasive maneuver (e.g. course alteration) is related to the modeling of the avoidance action. Collision avoidance is in fact a dynamic process presented by a group of evasive maneuvers, i.e. evasive trajectory. Vessel trajectory modeling is to determine continuous evasive maneuvers in a varying collision situation. For optimization task, the trajectory can be treated as a most economical path between an origin and a destination while enabling safe navigation.

Conflict-free planning is a traffic control approach to conflict resolution. Studies in vessel routing and scheduling are reviewed. Vessel routing has been developed to eliminate collision risk with safe voyage planning. The idea of routing is to allocate a route for each vessel so that encounter rate of all vessels is minimal in certain areas. However application of routing methods is restricted in port waters where vessels are specified with given routes and may not freely choose alternatives.

Studies in vessel scheduling are mostly developed for optimizing shipping problems. A few works are addressed for marine traffic scheduling. Scheduling represents a feasible approach to conflict free planning for port traffic. It is based on a general rule that, a vessel before arriving is required to report to the seaport its route, arrival time, and vessel characteristics. Although fluctuations may occur in real time, schedules for vessels can be preplanned according to the given information, so as to eliminate most of possible conflicts in advance.

2.3.3 Limitation and gaps in literature

A review of previous studies reveals that systematic methods for marine traffic conflict detection and resolution have not been found in literature. Almost all past studies are related to marine traffic concern with collision accidents, which implies the importance of navigational safety. A conflict can be considered as a collision risk with a low degree of danger. In this sense, studies relevant to collision risk detection and collision avoidance are reviewed as necessary references.

As reviewed in collision risk detection, ship domain is a comprehensive and accurate criterion for determination of a collision risk as well as a conflict. Conflict detection algorithms will be developed in Chapter 4 with use of the ship domain criterion.

Studies of collision avoidance provide evasive strategies for resolving real-time collisions/conflicts. On the other hand, collision avoidance focuses safety of the own vessel in direct control of navigator, but does not pay attention to the behavior of the whole network traffic. In this research, conflict resolution is considered as a traffic control problem which can be solved through planning and scheduling methods. However, relevant studies have not been developed for this purpose. Against the background, this research will make efforts on conflict-free planning by original algorithms and models. It can be deemed as a new direction in the field of marine traffic control.

Based on the literature review, the design of methodology will be presented in the following chapter.

Chapter 3
Methodological design

As discussed in Chapter 1, this research aims to propose feasible measures for port operators so as to resolve traffic conflicts and congestion within a seaport. In this chapter the methodology for systematically solving the proposed research problem will be presented. The methodology framework is consisting of a two stage planning strategy for conflict resolution and domain-based algorithms for conflict detection. Simulation approach will be applied for demonstrating and evaluating the measures of conflict detection and resolution. The overall design of the proposed simulation system will be described as well.

Development of the methodology in Chapter 3 is based on understanding the limitations and gaps in previous studies addressed in Chapter 2. This Chapter is focused on a whole framework of methodology design, e.g. why the two-stage conflict resolution strategy is adopted and how to implement them. Chapter 3 can be treated as guideline for the following Chapters 4, 5 and 6, where the details on method development will be presented.

3.1 Measures of conflict resolution

A conflict is perceived as a collision risk based on the consideration that the probability of a collision is dependent on the encounter rate of vessels (Macduff 1974). In previous studies, traffic conflicts as an alternative to collision risks are adapted in risk assessment for modeling and evaluating of collision risks (Pietrzykowski 1999; Pietrzykowski and Gucma 2002; Pietrzykowski 2008; Debnath *et al.* 2011; Weng, Meng *et al.* 2012).

As reviewed in Chapter 2, collision avoidance technology is to focus on real-time evasive actions of the involved vessels faced with an impending collision risk. In such case, navigational safety of the vessels involved is of the upmost and the only concern. It should be noticed that, besides safety concerns, conflicts also have significant influence on traffic congestion and delay which greatly affect efficiency of port operations. The significance of conflict resolution in traffic congestion management is of equal importance to its concerns on navigational safety. A review of past literature reveals that no systematic method has been developed for resolving conflicts in this regard.

This research aims to suggest/provide measures to resolve conflicts for planning and scheduling of port operators so as to enable minimize delay and congestion ahead of time (for example, in the next day). There are two aspects to be considered in conflict resolution. Safe navigation of vessels without conflicts is taken as the first priority. The second issue is concerned with reducing the impact of conflicts through minimizing total delay in the traffic network. Avoidance of real-time conflicts is not the scope of this research. The focus of this research is on enabling good traffic conditions within the entire traffic network, not merely improving that on an individual network element.

It is a traffic control problem to coordinate vessels without conflicts within a seaport. Conflict-free traffic is solved as a scheduling problem in airport ground traffic control and train station traffic control (Tomlin, Pappas *et al.* 1998; Menon *et al.* 1999; Pallottino, Feron *et al.* 2002; Pallottino *et al.* 2007). However, to the best of this researcher' knowledge, studies for conflict-free vessel scheduling cannot be found in literature. Scheduling methods are inadaptable to open sea where traffic in random order is hardly controlled. On the contrary, port water is characterized by a restricted zone where vessels move along given routes and schedules. The shore VTS system enables information transmission between traffic controller and vessels, which provides necessary condition for traffic planning. It is practical for port traffic controller to execute a preplanning process ensuring that vessels have safe and conflict-free schedules.

On the other hand, conflict resolution is not a simple scheduling problem which can be resolved in such a deterministic manner. Real vessel movements are stochastic in nature due to random fluctuations in vessel speed, and deviation from original schedules. The stochastic parts might result in unexpected conflicts which cannot be eliminated in the preplanning process. In this sense, we propose a systematic strategy for conflict resolution, which is implemented in two stages.

In the first stage, a preplanning process is executed for the planning of safe vessel movements before vessels actually arrive at a seaport. It can be referred to as vessel rescheduling by the port control center based on a given original schedule of arriving and departing vessels for the next day as reported by vessel operators. Most previous studies in vessel scheduling are developed for optimal tasks which resemble the routing problem (David 1983; Lokuge and Alahakoon 2007; Ulusçu, Özbaş *et al.* 2009), but few of them are interested in the safety of navigation. The conflict-free vessel scheduling problem is first discussed in this research. Iterative scheduling algorithms are developed here for coordinating vessels without conflicts for a given schedule.

Adjustments on vessel schedules bring delays to the vessels which require changes to the original schedule of vessels. To avoid congestion, these delays would, desirably, be absorbed enroute (for arrivals) or at the berths/anchorages (for departures) rather than within the traffic network. For example, vessel operators can be advised to postpone the arrival of their vessels so as to avoid delays to accumulate within the port traffic network. Nonetheless, it does not mean that vessels can wait outside the port for very long time. It is assumed that each vessel has a maximum acceptable delay (i.e. slack time) in its original schedule. Optimization is applied in the preplanning process, so that the total delay can be minimized subject to restrictions of maximum acceptable delay for each vessel.

Preplanning is a preprocessing stage for eliminating potential conflicts resulting from the original schedules of vessels. Nevertheless, it is unlikely to eliminate all conflicts that would occur in real-time vessel movements due to random fluctuations in vessel speed and deviations in timetables. The random fluctuations may result in potential conflicts. Therefore, the second stage of conflict resolution is to provide proper strategies for resolving such conflicts.

This stage is a simulation of vessel movements in the next day as a result of the stochastic events. For an individual vessel, the most direct way to resolve a conflict is to take evasive maneuvers. However, few port authorities actually control vessel maneuvers once the vessel's routing has been approved. It is hardly possible, in a real encounter, to consider so many variables with expert knowledge to come up with an optimal solution for evasive maneuvers.

In the sense of traffic control, a necessary measure is to avoid imminent conflict situations which are urgent events or emergencies. A multi-link, look ahead conflict prediction will be executed a few links in advance, thus allowing sufficient time for navigators to take evasive actions. The applicable corrective measures are identified by a decision-making mechanism. These measures conclude general rules or guidance for navigator chooses proper evasive maneuvers, but how the actual maneuvers are to be conducted is not of significance in this regard.

Evasive actions would result in delays to vessel operations. Such delays, compared to the delays incurred in the preplanning stage, arise from a dynamic traffic environment which will affect the efficiency of the traffic network. Traffic congestion will appear if delays are accumulated beyond the limitation of tolerable delay in a certain area. Therefore, the decision-making process will check and evaluate that there is sufficient space and time to perform corrective maneuvers while the total delay incurred within the affected area is acceptable. The basic idea of the decision-making process is based on Recognition-primed decision (RPD) strategy: looking back and looking forward, which simulates a naturalistic decision making process (Klein 1993; Chauvin and Lardjane 2008).

3.2 Methods of conflict detection

Implementation of conflict resolution is dependent on whether conflicts can be effectively detected. Conflict detection is another main objective of this research, which serves as the basis for conflict resolution. Methods of conflict detection are developed in two parts: conflict determination and conflict prediction.

A conflict is treated as a collision risk with a low degree of danger. Two criteria haven been used in previous studies for measuring collision risk: the closest point of approach (CPA) and ship domain. Parametric analysis with CPA applies to the open sea where vessels can freely take actions (Lisowski and Smierzchalski 1995; Lenart 1999; Lamb *et al.* 2000; Shi *et al.* 2007; Chauvin and Lardjane 2008), but is not so appropriate in restricted waters such as narrow fairways. In view of this, ship domain has been proposed as a more comprehensive and accurate criterion to measure collision risks (Fujii and Tanaka 1971; Goodwin 1975; Coldwell 1983; Pietrzykowski 1999; Smierzchalski and Michalewicz 2000; Szlapczynski 2006). In a port traffic system, a vessel traveling in fairways is required to keep safe clearances in accordance with the port’s regulation. The domain of a vessel can therefore be considered as the clearance area around it. The event of domain interference indicates a conflict between two vessels.

A conflict can be determined from the relative positions between the domains of two vessels. If the domain of one vessel will interfere (i.e. overlap) with the domain of the other vessel, a potential conflict is indicated. This method requires complicated and time-consuming calculations. In the preplanning stage, iterative rescheduling requires a fast and more efficient method for conflict determination. This is particularly true since, in a large number of cases, the relative positions of a pair of domains do not overlap. A study in aircraft movement simulation (Fan 1988) provides a reference of conflict-free planning. We therefore develop a conflict determination method on the basis of minimum time separation requirement between successive vessels. This is not precise detection but it can produce quick estimates of potential conflicts under the given vessel schedules.

In the simulation stage, random fluctuations in variables related to vessel movements may result in conflicts which need to be resolved. Evasive maneuvers by one or both vessels would be needed for conflict avoidance. In an urgent encounter situation, a vessel may not have sufficient time to take evasive action due to limited space or distance.

An algorithm is developed to predict vessel conflicts in a dynamic traffic environment. The criterion of ship domain is used for accurately measuring a conflict. That is, the relative positions of the domains of two vessels will be evaluated before they actually encounter each other. In the traffic network a fairway consists of a group of links. It is likely that predicting vessel positions only one link ahead may be insufficient, particularly if the length of the link is short. Therefore, conflict prediction will be executed by estimating vessel positions a few links in advance (called here multi-link-ahead), thus allowing sufficient time for navigators to take evasive actions. With the multi-link-ahead conflict prediction, potential conflicts can be resolved much earlier and are therefore unlikely to be as urgent collision situations. The development of this algorithm will be discussed in the following chapter.

3.3 Design of a simulation system

The main methodology of this research is a two-stage traffic planning strategy for resolving vessel conflicts. Conflict detection is another element in the methodological framework, which serves as the basis for conflict resolution. Conflict detection and resolution presents the issue related to method design and development. In this thesis, the other issue will be addressed for the purpose of method evaluation and demonstration.

Traditional mathematical analysis approach, e.g. risk assessment can be used for evaluating the measures of conflict detection and resolution through a quantitative analytical model. But, the mathematical analysis approach is not proper for this research to achieve the expected aims. Firstly, an analytical model requires a substantial amount of sound data for statistical modeling. The accuracy of risk assessment relies on the correctness and effectiveness of the data collected. This research has difficulty in accessing data maintained by external organizations such as port authorities without their cooperation, because quite a lot of the seaport information needed is confidential and sensitive and are protected. In addition, an analytical model is usually developed based on statistical characteristics of a certain traffic pattern, e.g. the Port of Singapore. It is noted that in this thesis the measures for conflict detection and resolution are addressed for a generalized traffic system in a seaport. An analytical model cannot satisfy the requirement for evaluating such general measures.

Computer simulation approach will be used as alternative. A computer simulation is an attempt to model either a real life or hypothetical situation on a computer so that it can be studied to see how the system works. Simulation is often useful in dealing with the following cases (Zeigler *et al.* 1976):

- The subject is very complex with many variables and interacting components;
- There are no other feasible solutions, or such other solutions are so complicated that they are difficult to implement;
- The study output is to be visual as in computer animation.

Use of computer simulation has numbers of advantages for this research. Firstly, the simulation approach has good efficiency in integrating complex systems (Prähofer *et al.* 2000).The traffic system within a seaport consists of many variables and interacting components. The simulation tool gives the designer the capability to integrate data, models and resolutions with modularity and flexibility (Hassan 1993). Moreover, simulation is a useful adjunct and an effective assistant to mathematical methods. It provides an experiential platform for testing the applicability of conflict detection and resolution methods in a certain traffic environment before executing them in practice. In particular simulation technology has good performance in computer animation which presents a direct display of dynamic vessel movements and traffic environment. With simulation technology, conflict scenarios that are displayed visually could aid traffic controller to investigate serious traffic incidents/accidents without involving field trials. Admittedly simulation also would require accurate data to provide realistic results, but it is a more flexible approach which is not strictly sensitive to the data required and allows one to investigate serious problems that may result, even the input data needs to be further refined.

Marine traffic simulation has been conducted in previous studies for two purposes:

- Existing simulation tools or develop simulators are used to conduct trials for verifying the developed algorithms or models (Smierzchalski and Michalewicz 2000; van Hilten *et al.* 2000; Chang, Jan *et al.* 2003; Pedersen, Inoue *et al.* 2003; Zhang *et al.* 2009);

- Develop a simulation model for a traffic system, so as to investigate or evaluate the performance of traffic activities as a whole or on certain system elements (Hasegawa *et al.* 2001; Kose *et al.* 2003; Merrick, Dinesh *et al.* 2003; Hasegawa 2004; Ince *et al.* 2004).

This research will focus on a simulation study of marine traffic conflict within a seaport, which can be treated as a combination of the above two kinds of studies. On one hand, it provides an experimental platform for evaluating the measures for conflict detection and resolution. On the other hand, it as well is a basis for implementing the real time conflict detection and resolution in a traffic system. An application of this simulation model will be demonstrated using the seaport of Singapore as an example. Inputs data include the background map of Singapore seaport, data on fairways, and information on vessel types and characteristics. Vessel arrivals and vessel routes are generated by the simulation system according to real data. It is emphasized here that the simulation model is not developed for a specific traffic system. It is a generic model which can be adapted to other busy seaports that are faced with conflict risks and traffic congestion. Simulation can be used as an experimental platform where data required can be substituted by parameters thus be simplified as basic input.

The current simulation system developed in this research is an offline system for the traffic control center executes planning before vessel arriving to the seaport. An online simulation system can be developed to extend the work into real-time conflict detection and resolution, to point out possible directions of further work.

An integrated multifunctional simulation system, called Marine Traffic Conflict Simulation System, is proposed. The simulation system is developed by Visual C++ under Windows XP environment. Design of this simulation system takes the following considerations into account:

1) It should be applicable in a variety of water areas;
2) Complicated conflict scenarios (e.g. multi-vessel conflict) can be investigated;
3) The process of simulation is displayed graphically;
4) A user interface which allows people to interact with the simulation computers**;**
5) Data transmission and communication;
6) Good compatibility and expandability with other platforms.

Figure 3.1 depicts the framework of this simulation system. The details for development of the simulation system will be discussed in Chapter 6.

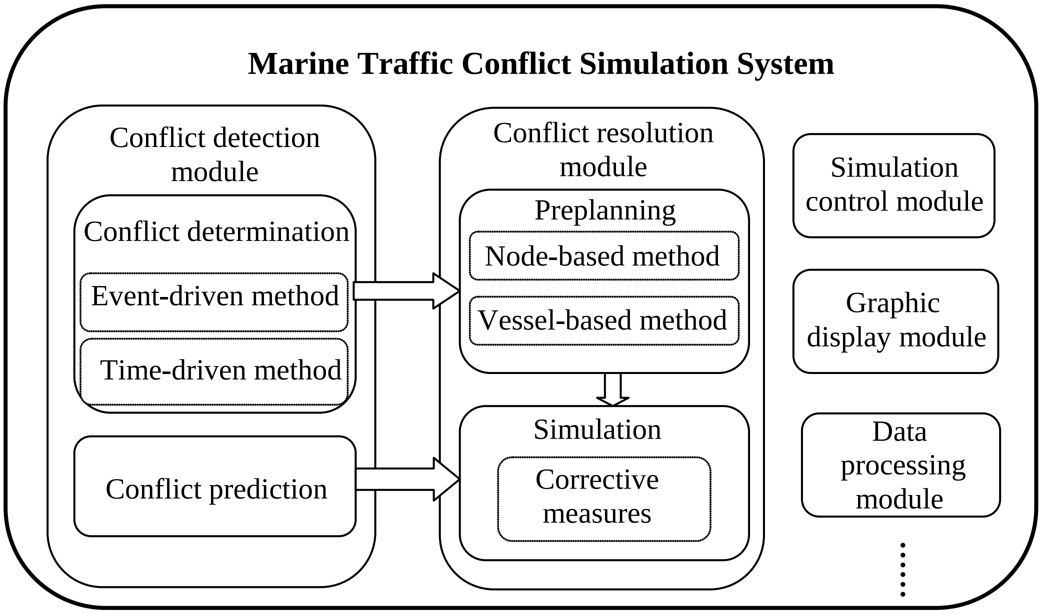


Figure 3.1 Framework of the simulation system.

3.4 Summary

This Chapter is focused on design of the methodology for detecting and resolving conflicts within a seaport.

A two-stage strategy for resolving traffic conflicts within a seaport is presented. The first stage is the preplanning which will detect and resolve potential conflicts in a deterministic manner using the original schedule of vessel arrivals and departures. The second stage is to take care of the stochastic part due to random fluctuations in real vessel movements. The resolution is to simulate vessel movements in the next day and identify applicable corrective measures with multi-link look-ahead conflict prediction. The two stages use different technologies for minimizing delays and reducing congestion in the traffic lanes.

Methods for conflict detection are designed with the use of ship domain criterion. Conflict determination will be conducted in the preplanning stage for detecting existing conflicts under original vessel schedules. For the simulation stage, conflict prediction can forecast conflicts multi-link ahead current vessel position, so as to avoid urgent encounter and allow the investigation of corrective measures for conflict resolution.

Development of the simulation system is to provide a platform/basis for evaluating the strategy of conflict detection and resolution. Simulation is a tool for investigating conflict scenarios with good performance in computer animation and a direct display of dynamic vessel movements and traffic environment. This research will focus on a simulation study, for marine traffic conflict within a general seaport characterized by a traffic network consisting of basic elements, e.g. fairways, anchorages and berths etc. An application of the simulation system is demonstrated using the seaport of Singapore as an example, which will be discussed in Chapter 7.

Chapter 4

Conflict detection[1]

Conflict detection can be treated as the process of determining when a conflict between vessels will occur, which severs as the basis of conflict resolution. As discussed in Chapter 2, parametric analysis of CPAs is difficult to use in restricted waters such as narrow fairways. Alternatively ship domain is proposed as a more comprehensive and accurate criterion for measuring a conflict. A conflict is referred to as an event of interference between the two vessels' domains. It is possible to detect a conflict through evaluating the relative positions of the two domains. This chapter presents the algorithms to determine and predict conflicts using the criterion of ship domain.

The conflict detection methods developed in this chapter will be used in conflict resolution in Chapter 5. It is also a function module of the simulation system proposed in Chapter 7.

---

[1] The following publications are based on the results of this chapter:

Li, Q. and Fan H.S.L. (2012). "A Simulation Model for Detecting Vessel Conflicts Within a Seaport", International Journal on Marine Navigation and Safety of Sea Transportation 6(1): 11-17.

Li, Q., Lam J.L.S., and Fan H.S.L. (2012) "Multi-link-ahead Conflicts Prediction in Dynamic Seaport Environments", Serious Games and Simulation.

4.1 Preliminaries

A seaport traffic system can be treated as a network, which is a schematic representation of traffic fairways and their intersections in port waters. This section defines key elements and rules for modeling of vessels and the traffic network, which will be used in the rest of this thesis.

4.1.1 Notations of a vessel

A vessel is denoted as $V(O, D, W, L, \bar{W}, \bar{L}^1, \bar{L}^2)$. As shown in Figure 4.1, the shape of the vessel is simplified as a rectangle P whose dimensions are W (width) and L (length). Suppose $O(x, y)$ is the center of the rectangle P. At present the vessel is traveling along the direction D.

The shape and size of ship domain are affected by factors such as vessel speed and dimensions, location in the sea space network, and traffic density. As different factors are considered, ship domains proposed by various studies differ from one another (Davis, Dove *et al.* 1980; Coldwell 1983; Zhu, Xu *et al.* 2001; Pietrzykowski 2008). In a seaport traffic system, vessels traveling along fairways are required to keep various safe clearances in accordance with the port's regulations. The domain of a vessel can be referred to as the clearance area around it, within which the vessel can keep safe distance to avoid conflicting with others. As shown in Figure 4.1, the clearance area is represented by a rectangular zone, denoted as $Q(\bar{W}, \bar{L}^1, \bar{L}^2)$. The parameter $\bar{W}$ represents the lateral clearance, while $\bar{L}^1$ and $\bar{L}^2$ represent the longitudinal clearances in the direction of the bow and in the direction of the stern, respectively.

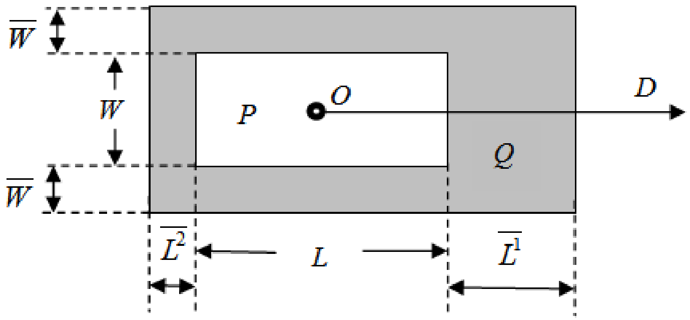


Figure 4.1 A vessel and its domain.

4.1.2 Representation of the traffic network

A seaport traffic system can be treated as a network consisting of nodes and links (Fan and Cao 2000) (Figure 4.2). Within the network each link indicates a section of a fairway, and a node can be a berthing/anchorage area, a boarding point for port pilots, an intersection area of fairways, or a separation point dividing a fairway into two sections due to differences in widths and/or traffic regulations. The separation nodes do not exist in real life, but are developed for the purpose of modeling.

Figure 4.2 shows an example of the traffic network of the Port of Singapore. As displayed in this network, nodes are simplified as dots regardless of actual dimensions. Here, green dots refer to boarding points; blue dots represent anchorage areas; red dots are berthing areas; and black dots refer to separation points and intersections of fairways. Links are represented by grey rectangle areas. Each link is specified with certain attributions on fairway features, e.g. length, width, numbers of lanes, and regulations such as speed limitation, maximum acceptable delay and so on.

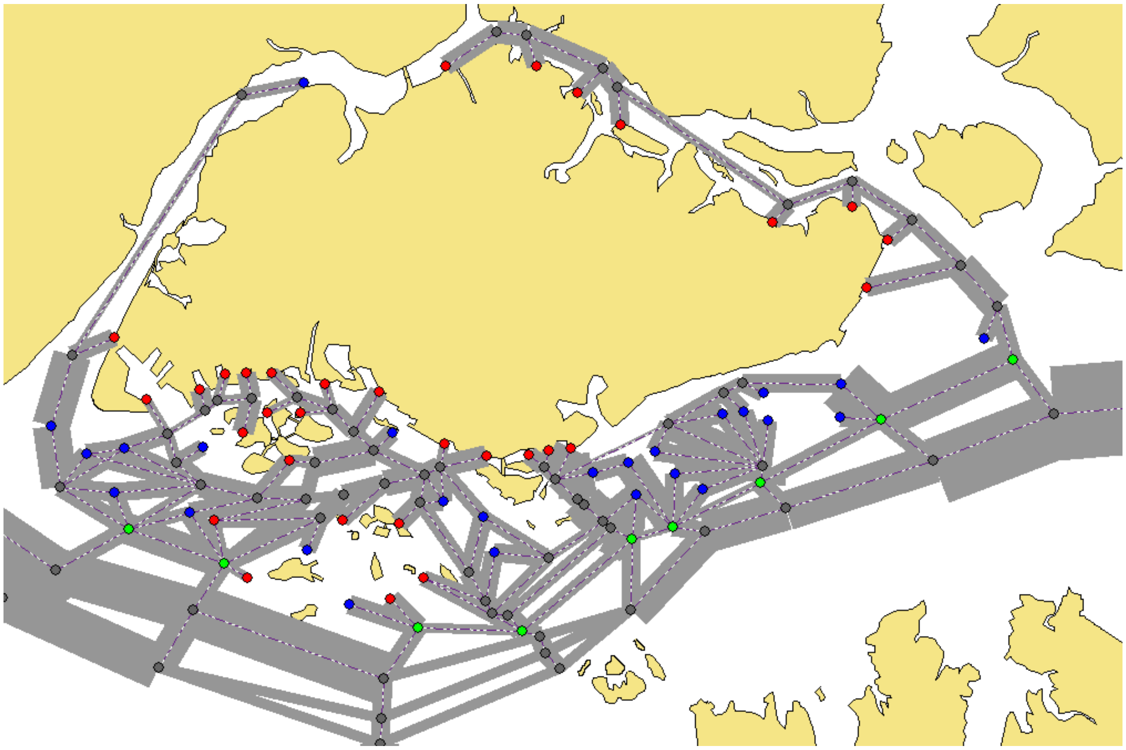


Figure 4.2 Traffic network for Port of Singapore.

Key: green dots - boarding points; blue dots - anchorage areas; red dots - berthing areas; black dots - separation points and intersections of fairways; grey rectangle areas – links.

4.1.3 Modeling of vessel path

The simulation system requires a vessel path to be assigned so as to track the position of a vessel within the traffic network. This path refers to a vessel trajectory based on basic navigational maneuvers, i.e. straight-line course-based trajectories in links and turning-maneuver-based trajectories along curves.

Vessel trajectory based on straight-line course is not strictly straight but is affected by wind and tide. There is much uncertainty to determine vessel varying motion by taking into account the external disturbances. A reasonable simplification supposes that a vessel keeps a straight line along the centerline of the traffic lane. As shown in Figure 4.3, vessel trajectories in a two-way traffic link are indicated by the red lines.

A vessel should maintain a continuous and smooth movement during a turning process. Maneuvering of a constant radius turn enables a steady turning with less drift angle and less speed loss, which is commonly applied in marine navigation and piloting (Aarsaether *et al.* 2007). A method is proposed here to model vessel trajectory based on the constant radius turning maneuver.

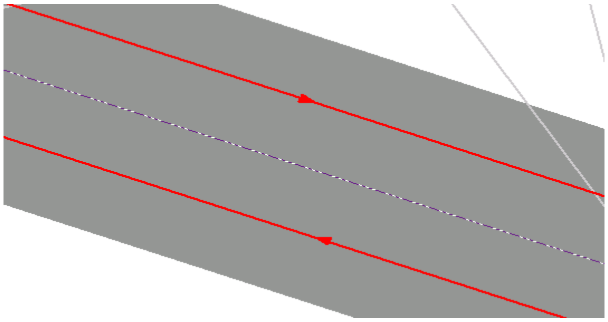


Figure 4.3 Vessel trajectories in a link.

As is shown in Figure 4.4, a vessel is traveling along the link c_1 and moving towards the link c_2., Without considering the time of rudder reaction, the turning trajectory of the vessel will follow a circular arc under a constant radius turn. It is possible to determine the arc given the bend speed and rate of turn (ROT).

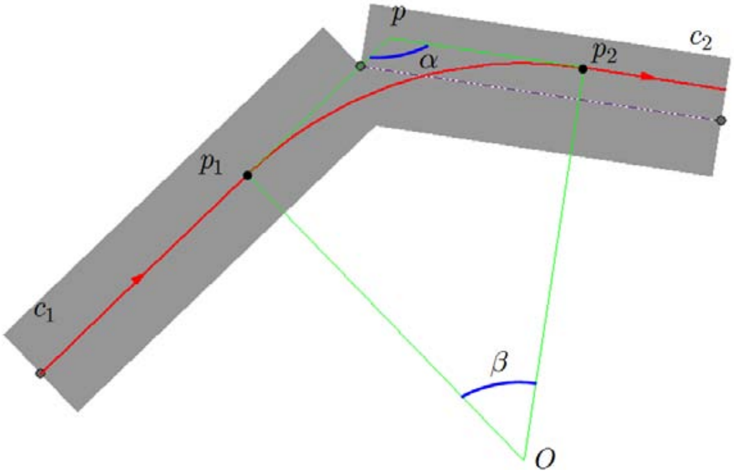


Figure 4.4 Vessel crossing trajectory in a junction.

Suppose that the angle between the two links is α (degree). Bend speed and ROT are given as vessel parameters. Geometrical analysis has the following equations,

$$\begin{aligned}
&\beta = 180 - \alpha, \\
&t = \beta / r, \\
&L = 60vt, \qquad\qquad (4.1) \\
&R = |op_1| = |op_2| = 360L/2\pi\beta, \\
&D = |p_1 p| = |p_2 p| = R \cdot \tan(\beta / 2),
\end{aligned}$$

where

β: the angle at which the vessel needs to turn (degree),

t: the time required for the vessel to cross the junction,

r: rate of turn,

L: the length of the required arc,

v: the bend speed,

R: the radius of the required arc,

o: the center of the required arc,

p: the intersection point of links c_1 and c_2,

$p_{1:}$ the point on link c_1 where the vessel starts turning,

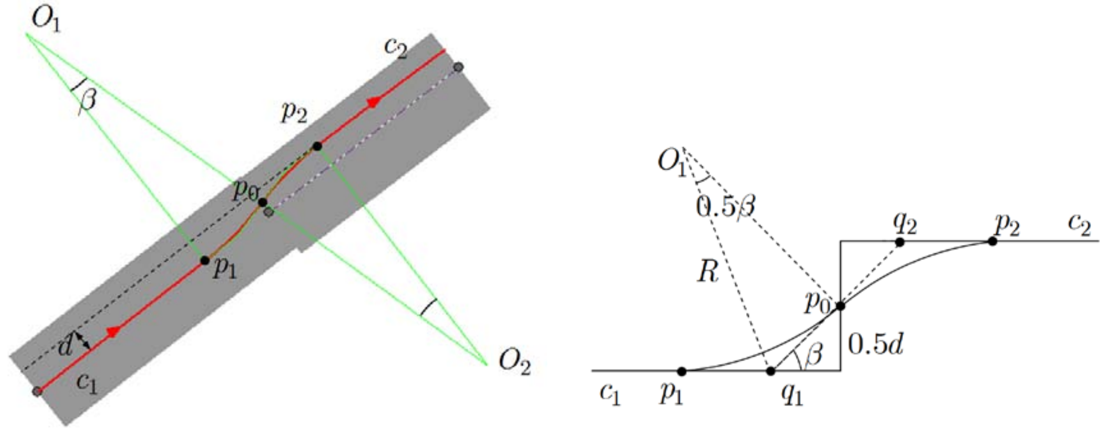
$p_{2:}$ the point on link c_2 where the vessel ends turning,

D: the distance from p_1 to p (or p to p_2).

Eq.(4.1) is used to calculate the points p_1 and p_2. The circular arc can be determined as long as the turning points are known.

Sometimes a vessel needs to change course (i.e. turn) twice across two links (see Figure 4.5). A similar method is used for determining vessel turning trajectory. In Figure 4.5(a), a vessel is traveling across link c_1 to c_2. Its turning trajectory is an S-shaped curve which is supposed to be two same circular arcs, denoted as A_1 and A_2. Let p_1, p_2 and p_0 be the turning points, which means the vessel will make the first turn at p_1 to p_0, and then it will make the second turn at p_0 to p_2. The turning points satisfy the following conditions,

- A_1 is tangent to link c_1 at point p_1,
- A_2 is tangent to link c_2 at point p_2, and
- A_1 is tangent to A_2 at point p_0.



(a) (b)

Figure 4.5 Vessel crossing trajectory between two parallel links.

To calculate the locations of these turning points, assume that,

- The distance between links c_1 and c_2 is d.
- The crossing trajectory is a centrosymmetric S-shaped curve whose center is p_0 (Figure 4.5(b)). The turning angle that the vessel makes for each turn is β. The radius of each turn is R, which can be calculated from Eq.(4.1).
- q_1, q_2 are two points on links c_1, c_2, respectively, which satisfy the condition that line $\overline{q_1 p_0 q_2}$ is tangent to both A_1 and A_2 (Figure 4.5(b)).

In Figure 4.5(b), the following equation holds

$$|p_0 q_1| = d/(2\tan\beta) = R \cdot \tan(\beta/2). \tag{4.2}$$

Thus,

$$\beta = 2\arcsin\sqrt{d/(4R)}. \tag{4.3}$$

Supposing $L = |p_0 q_1|$ leads to the following equation,

$$L = |p_1 q_1| = |p_0 q_1| = |p_0 q_2| = |p_2 q_2| = R \cdot \tan(\beta/2). \tag{4.4}$$

The turning points p_1, p_0 and p_2 can be calculated using Eq.(4.3) and Eq.(4.4). These then determine the vessel's crossing trajectory.

In most cases, c_1 and c_2 are not so far apart that a vessel would make turns of a small angle for each turn, i.e. β is less than 90°. This implies

$$\beta = \arcsin\sqrt{d/(4R)} < \pi/4 \Rightarrow \sqrt{d/(4R)} < \sqrt{2}/2 \Rightarrow d < 2R.$$

When links c_1 and c_2 are far apart, i.e. $d >= 2R$, a vessel cannot cross through the two links by making two consecutive turns of 90° each. In this case, the vessel has to travel an additional straight path between the two turns. Figure 4.6 shows such a case in which the vessel's crossing trajectory consists of three parts: two circular arcs (i.e. A_1 and A_2) and a straight line (i.e. $\overline{r_1 r_2}$). This leads to the following equations,

$$\begin{aligned} &\beta = \pi/4, \\ &| p_1 o_1 | = | o_1 r_1 | = | r_2 o_2 | = | o_2 p_2 | = R, \end{aligned} \tag{4.5}$$

where

p_1, r_1, p_2, r_2: vessel turning points,

o_1, o_2: the centers of arc A_1 and arc A_2.

Eq.(4.5) can be used to calculate the locations of p_1, r_1, p_2 and r_2. These then determine the vessel's crossing trajectory which is shown in Figure 4.6.

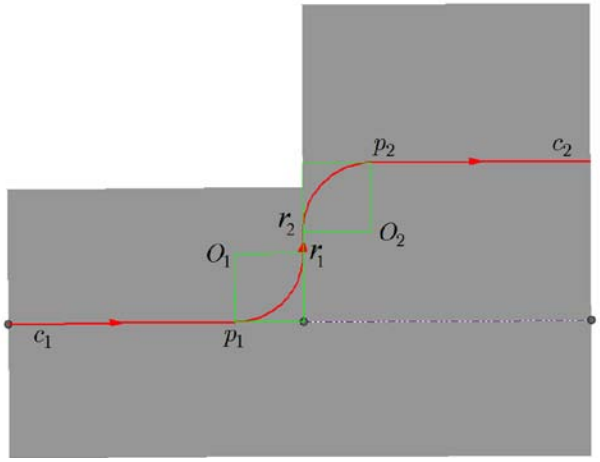


Figure 4.6 Vessel crossing trajectory with $D > 2R$.

4.2 Algorithm design

A conflict is defined as an event of domain interference between two vessels. Conflict detection algorithm developed in this study comprises two parts: conflict determination and conflict prediction. Conflict determination aims to detect occurrence of the event of domain interference through evaluation of positional relationship between two domains. Conflict determination can be performed to detect conflicts existing in the original schedules. Conflict prediction focuses on forecasting the risk before a conflict, i.e. the event of domain interference occurs. Multi-link-ahead conflict prediction enables sufficient time for vessels to take actions before a close conflict encounter while keep delay low.

4.2.1 Conflict determination

Let Q_1 and Q_2 be the domains of the two vessels V_1 and V_2, respectively. Mathematically, the conflict occurs if and only if Q_1 and Q_2 intersect with each other. As shown in Figure 4.7, a conflict occurs between V_1 and V_2.

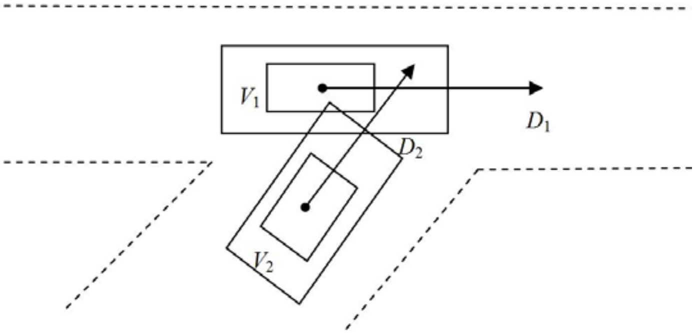


Figure 4.7 A conflict situation.

No conflict will exist if no part of a vessel's domain overlaps any part of the other vessel's domain. Otherwise, a conflict will be detected. In a special case, a rectangle may be enclosed by the other rectangle (e.g. Q_1 is enclosed by Q_2), which is also a conflict situation.

There are three positional relationships between a vessel's domain and another vessel's domain: separated, intersected and enclosed. Either of the latter two implies a conflict. Conflict determination is to evaluate the relationship between two rectangles, i.e. domains.

However, it may be time-consuming to evaluate the relationship between two rectangular domains among all vessels in the seaport. An algorithm has to run $C_N^2 = \frac{(N-1)(N-2)}{2}$ times for N vessels. In real life, most vessels are far away from each other, and conflicts are rare events. A preprocessing method is described below to eliminate the cases where vessels are far away from each other.

As shown in Figure 4.8, the circle represents a vessel's clearance area, whose center is c and radius is r. In an encounter situation, if the distance between a link c_1 and another link c_2 is larger than the sum of r_1 and r_2, the two circles are separated. It means no conflict will occur. Otherwise, one would analyze further to determine if a conflict will occur through an evaluation of the relationship between rectangles. This preprocessing greatly reduces computations due to a much simpler analysis between circles compared to that between rectangles. The process of determining whether two vessels are in a conflict is presented in a flowchart shown in Figure 4.9. It involves two steps:

(1) Evaluation of the relationship between two circles, and

(2) Evaluation of the relationship between two rectangles.

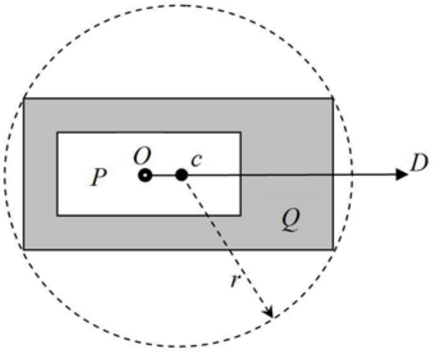


Figure 4.8 The clearance area of a vessel.

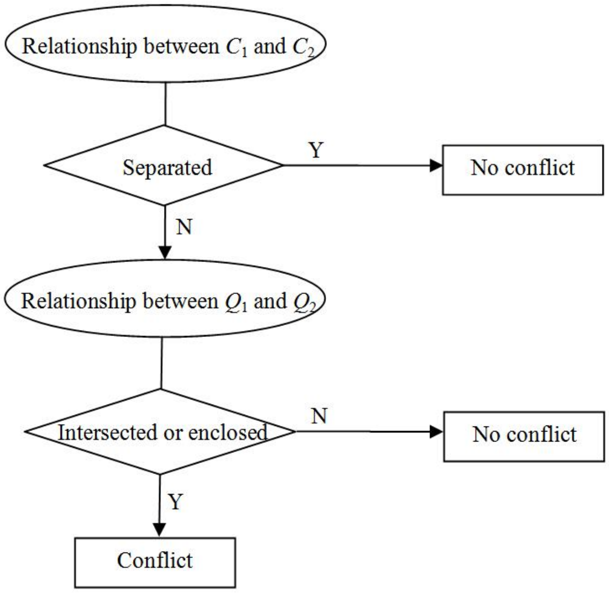


Figure 4.9 Flowchart of conflict determination.

Relationship between two circles

In Figure 4.10, C_1 (c_1, r_1) and C_2 (c_2, r_2) are the circumcircles of Q_1 and Q_2. When the distance between point c_1 (x_{c1}, y_{c1}) and point c_2 (x_{c2}, y_{c2}) is larger than the sum of r_1 and r_2, no conflict will occur. Referring to notations in Figure 4.1, x_{c1}, y_{c1}, x_{c2}, and y_{c2} can be obtained accordingly. Thus, no conflict exists between the two vessels if

$$\left\| c_1 - c_2 \right\|^2 = \left(x_{c1} - x_{c2} \right)^2 + \left(y_{c1} - y_{c2} \right)^2 \geq \left(r_1 + r_2 \right)^2.$$

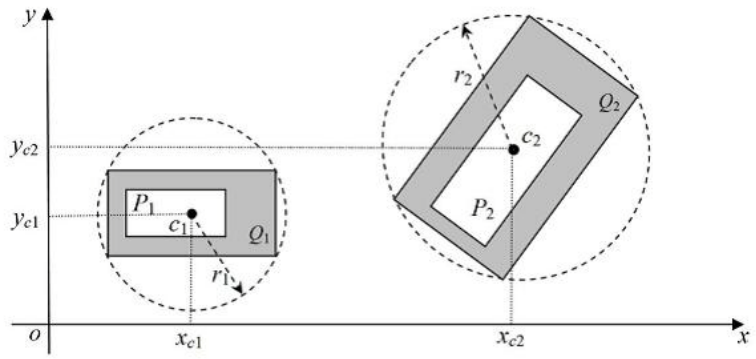


Figure 4.10 Relationship between two circles.

Relationship between two domains

There are three relationships between Q_1 and Q_2: separated, intersected and enclosed (see Figure 4.12). When the two rectangles interfere with each other, inclusive of the cases of intersecting, or one is enclosed in the other, a conflict occurs. In the case of a rectangle is enclosed in the other, there are two possibilities: Q_1 is enclosed in Q_2, or Q_2 is enclosed in Q_1. Both possibilities are taken into considerations. The flowchart to determine the domain relationship between two vessels is shown in Figure 4.11. The method for mathematical calculation of the three relationships is described in Appendix A.

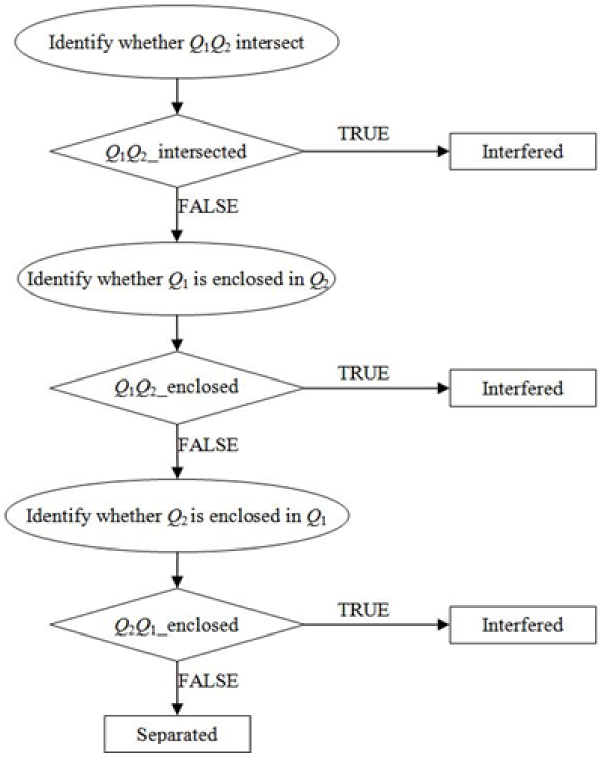


Figure 4.11 Flowchart of determining the relationship between two domains.

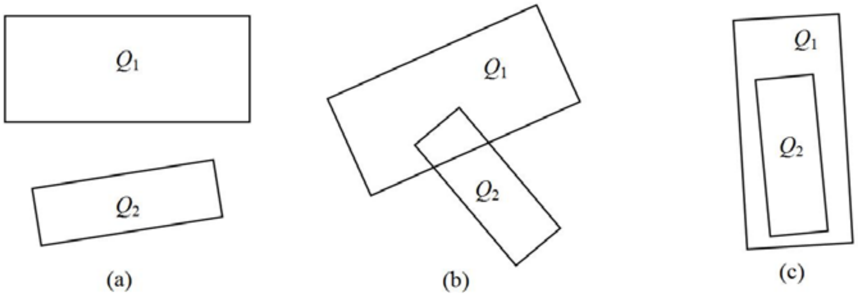


Figure 4.12 Three relationships of domains.
(a) separated, (b) intersected, and (c) enclosed.

4.2.2 Conflict prediction[2]

The idea of conflict prediction is to evaluate the relative positions of the domains of two vessels before they actually encounter. If the domain of a vessel will interfere with the domain of the other, a potential conflict is indicated.

There are two situations in conflict prediction:

- Node conflict prediction: a conflict occurs between two vessels when they cross through the same node.
- Link conflict prediction: a conflict occurs between two vessels when they travel along the same link.

In the first situation, the two vessels may have a conflict when they are crossing the same node. As discussed in Section 4.1.3, the trajectory when a vessel crossing a node is modeled by circular arcs. This simplification is for the purpose of predicting a conflict with the vessel position estimated in straight segments. A circular arc is simplified as a group of linked line segments each of which corresponds to the vessel's path for with a turning angle of 30^o. For example, the vessel trajectory in a crossing situation (see Figure 4.4) is represented by a group of linked lines colored in red in Figure 4.13. When two vessels move towards the node, the simulation system needs to predict whether a conflict will occur between them in each line segment.

In the second situation, the two vessels may encounter a conflict on a link. If the fairway is sufficiently wide so that one vessel can safely pass through the other, the conflict will not occur. Thus, link width should be considered in predicting conflicts along a link.

---

[2] The following publication is based on the results of this section:
Li, Q. and Fan H.S.L. (2012). "A Simulation Model for Detecting Vessel Conflicts Within a Seaport", International Journal on Marine Navigation and Safety of Sea Transportation 6(1): 11-17.

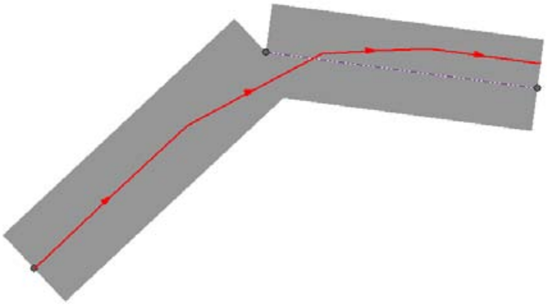


Figure 4.13 Vessel turning trajectory for conflict prediction.

4.2.2.1 Flowchart

The relationship between the positions of two vessels varies as the vessels move along their paths. The conflict situation would therefore change accordingly. Suppose the vessels have a risk of conflict during a certain time period. Based on the changes in vessel trajectories, this time period is divided into several time intervals. The system needs to separately evaluate the conflict situation during each time interval. Figure 4.14 shows the flowchart of conflict detection (noted that the notations (t_0, t_1, t_2, t_3) will be defined in Section 4.4.4.2 and Section 4.4.4.3).

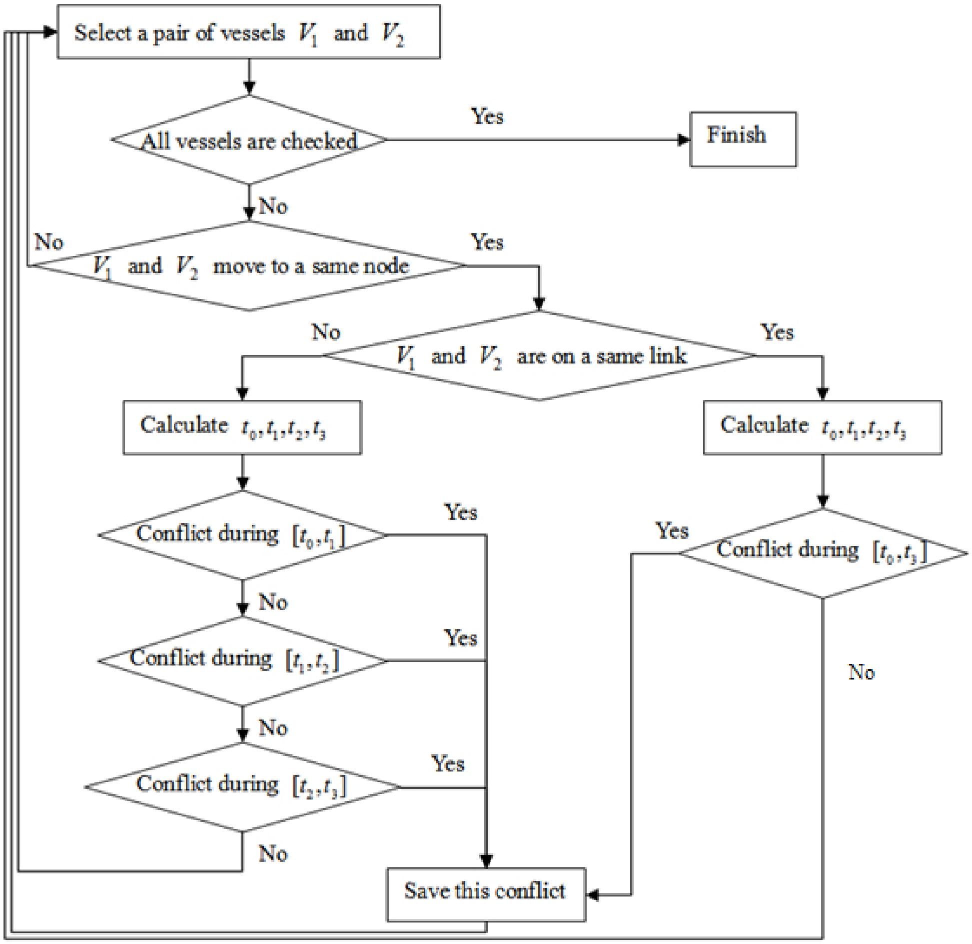


Figure 4.14 Flowchart of conflict prediction.

4.2.2.2 Predicting of conflict at a node

Two vessels, V_1 and V_2, on different links travel toward the same node. Table 4.1 lists the navigation information, where $t_1 < t_2$, i.e. V_1 will reach the node before V_2. Suppose $t_0 = 0, t_3 = \min(\bar{t}_1, \bar{t}_2)$. The target is to check whether there is any conflict during the time interval (0, t_3). The movements of V_1 with respect to V_2 are different in these three time intervals:

- In the time interval (t_0, t_1), the velocity of V_1 with respect to V_2 is $w_1 = v_1 - v_2$.
- In the time interval [t_1, t_2], the velocity of V_1 with respect to V_2 is $w_2 = \bar{v}_1 - v_2$.
- In the time interval (t_2, t_3), the velocity of V_1 with respect to V_2 is $w_3 = \bar{v}_1 - \bar{v}_2$.

Table 4.1 Vessel turning information.

	V_1	V_2
Velocity before the first turn	v_1	v_2
Velocity after the first turn	$\bar{v}_1$	$\bar{v}_2$
Time to the first turn	t_1	t_2
Time to the second turn	$\bar{t}_1$	$\bar{t}_2$

Figure 4.15 shows the movement of the center of V_1 relative to the center of V_2. With respect to V_2, starting at *A*, V_1 passes *B* at t_1 (see Figure 4.15(a)), moves from *B* to *C* during [t_1, t_2] (see Figure 4.15 (b)), and reaches *D* at t_3 (see Figure 4.15(c)). Thus,

$$\begin{aligned} AB &= w_1 t_2 = (v_1 - v_2) t_2, \\ BC &= w_2 (t_1 - t_2) = (\bar{v}_1 - v_2)(t_1 - t_2), \\ CD &= w_3 t_3 = (\bar{v}_1 - \bar{v}_2) t_3. \end{aligned}$$

At location *A*, the domain of V_1 follows its moving direction v_1 (Figure 4.16(a)). Similarly, the domains of the vessels at different locations can be obtained (see Table 4.2). Suppose that $q_{ij}^5 = q_{ij}^1$, *i* = 0,1, *j* = 0,1,2, *k* = 1,2,3,4, *i*=0,1. Table 4.2 tells that

- Q_{ij} is the domain of vessel V_i at $t = t_j$,
- q_{ij}^k is the k-th corner of the domain Q_{ij},
- $q_{ij}^k q_{ij}^{k+1}$ is the k-th edge of the domain Q_{ij}.

The movement of the domain of V_1 with respect to the domain of V_2 is denoted as the relative movement of V_1 to V_2. Figure 4.16 shows the relative movements of V_1 to V_2, in the three different time intervals depicted in Figure 4.15.

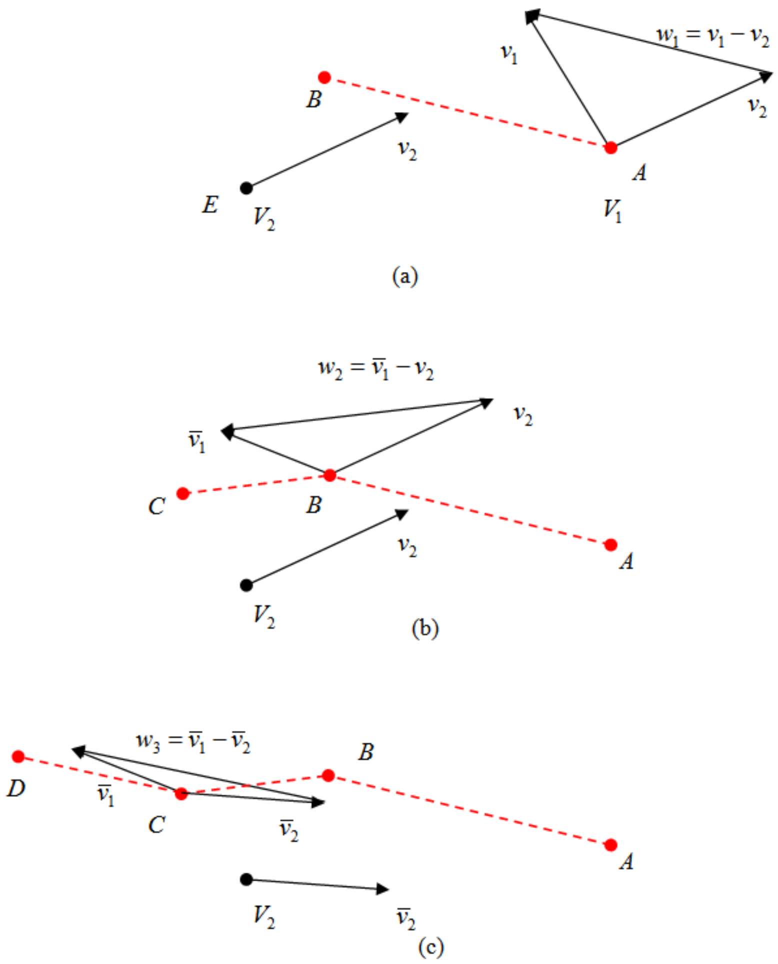


Figure 4.15 Movements of the center of V_1 with respect to the center of V_2. (a) in time interval $(0, t_1)$, (b) in time interval $[t_1, t_2]$, and (c) in time interval (t_2, t_3).

Table 4.2 Domains of vessels at different locations.

Location at time	Domain of V_1	Domain of V_2
$t = t_0$	$Q_{10}(q_{10}^1, q_{10}^2, q_{10}^3, q_{10}^4)$	$Q_{20}(q_{20}^1, q_{20}^2, q_{20}^3, q_{20}^4)$
$t = t_1$	$Q_{11}(q_{11}^1, q_{11}^2, q_{11}^3, q_{11}^4)$	$Q_{21}(q_{21}^1, q_{21}^2, q_{21}^3, q_{21}^4)$
$t = t_2$	$Q_{12}(q_{12}^1, q_{12}^2, q_{12}^3, q_{12}^4)$	$Q_{22}(q_{22}^1, q_{22}^2, q_{22}^3, q_{22}^4)$

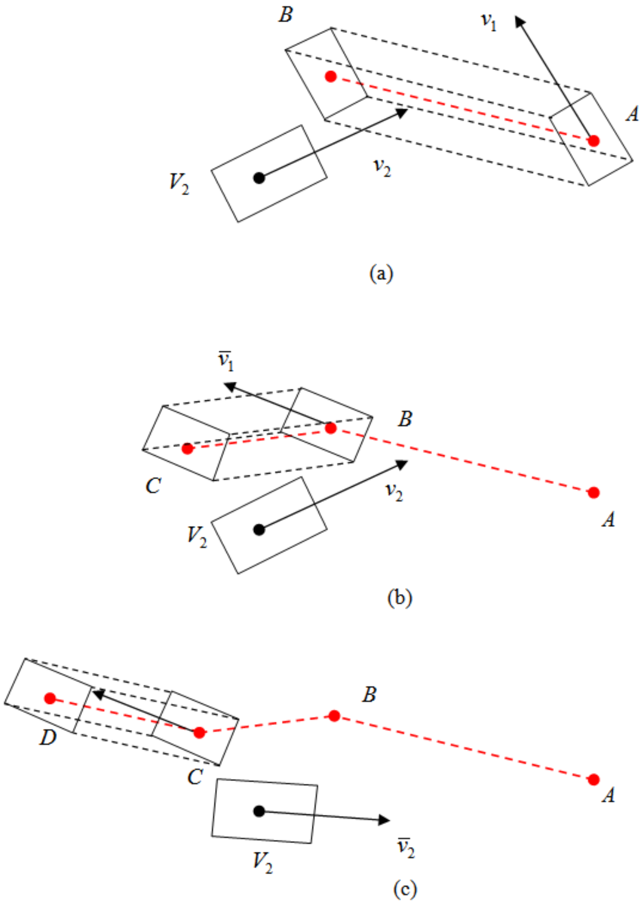


Figure 4.16 Relative movement of V_1 to V_2.

(a) in time interval $(0, t_1)$, (b) in time interval $[t_1, t_2]$, and (c) in time interval (t_2, t_3).

For any $j = 0, 1, 2$, in the time interval (t_j, t_{j+1}), the velocity of V_1 with respect to V_2 is w_{j+1}. The movement of the corner q_{1j}^k with respect to V_2 is a line segment $q_{1j}^k p_{1j}^k$ with

$$p_{1j}^k = q_{1j}^k + (t_{j+1} - t_j) \cdot w_{j+1}.$$

Thus, the movement of the edge $q_{1j}^k q_{1j}^{k+1}$ with respect to V_2 is $P_j^k = q_{1j}^k q_{1j}^{k+1} p_{1j}^{k+1} p_{1j}^k$ (Figure 4.17). If V_1 and V_2 conflict with each other, the movement of at least one edge of V_1 will intersect with the domain of V_2, i.e. $P_j^k \cap Q_{2j} \neq \varnothing$. Figure 4.17 shows an example when there is no conflict between V_1 and V_2. Figure 4.18 is another example when there is a conflict between V_1 and V_2.

In summary, V_1 and V_2 will conflict in the time interval (t_j, t_{j+1}) if and only if

$$\bigcup_{k=1}^{4}\left(P_j^k \cap Q_{2j}\right) \neq \varnothing .$$

In this way, conflict detection becomes a task of checking whether two parallelograms intersect with each other. The checking procedure can be implemented following the method in Appendix A.

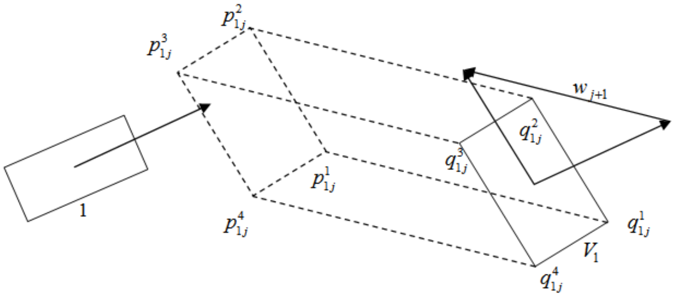


Figure 4.17 Since $P_j^k \cap Q_{2j} = \varnothing$, V_1 does not conflicts with V_2 in interval (t_j, t_{j+1}).

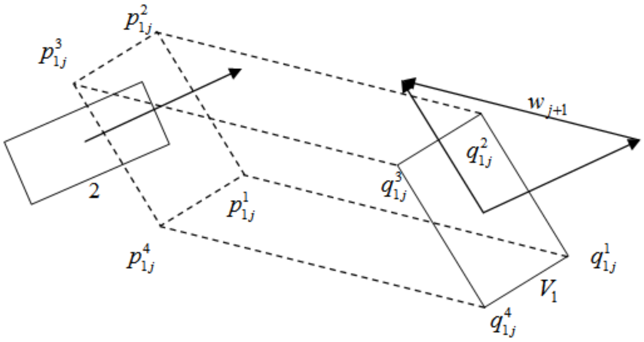


Figure 4.18 Since $P_j^2 \cap Q_{2j} \neq \varnothing, P_j^3 \cap Q_{2j} \neq \varnothing$, V_1 conflicts with V_2 in interval (t_j, t_{j+1}).

4.2.2.3 Predicting of conflict on a link

Suppose a vessel V_1 is following behind another vessel V_2 on a link (see Figure 4.19(a)). Table 4.3 lists the navigation information of the vessels. The velocity of V_1 with respect to V_2 is $w_1 = v_1 - v_2$. If $v_1 \leq v_2$, V_1 and V_2 will conflict if and only if $\left|AE\right| < \frac{L_1+L_2}{2}$.

Table 4.3 Information of two vessels travelling along the same link.

	V_1	V_2
Position	A	E
Velocity	v_1	v_2
Domain	$Q_1(q_1^1, q_1^2, q_1^3, q_1^4)$	$Q_2(q_2^1, q_2^2, q_2^3, q_2^4)$
Domain length	L_1	L_2
Domain width	W_1	W_2
Time to leave	t_1	t_2

Set $t_3 = \min(t_1, t_2)$. The simulation system needs to check whether the two vessels will conflict with each other during the time period (0, t_3). After that, the two vessels will not be conflicting on this link, because V_2 would have left this link.

If $v_1 > v_2$, the movement of V_1 relative to V_2 during (0, t_3) is shown in Figure 4.19(b), where $p_1^k = q_1^k + w_1 \cdot t_3, \quad k = 1, 2, 3, 4$. Obviously, V_1 and V_2 will conflict if and only if $q_1^1 q_1^4 p_1^2 p_1^3$ intersects with Q_2. In Figure 4.20(a), $q_1^1 q_1^4 p_1^2 p_1^3$ does intersect with Q_2, and hence V_1 and V_2 will conflict. In Figure 4.20(b), $q_1^1 q_1^4 p_1^2 p_1^3$ does not intersect with Q_2, therefore V_1 and V_2 will not conflict with each other.

(a) (b)

Figure 4.19 Vessels along the same link.
(a) V_1 follows V_2 along a link, and (b) the movement of V_1 relative to V_2 ($v_1 > v_2$).

Figure 4.20 A conflict on the same link depends on initial vessel positions.
(a) V_1 and V_2 will conflict, and (b) V_1 and V_2 will not conflict.

If the link is sufficiently wide, a vessel can overtake the other vessel without entering that vessel's domain. This overtaking situation is considered here. Suppose the width of the link is W, and the width of the domains of the two vessels are W_1 and W_2. The two vessels can travel in parallel without a conflict if $W \geq W_1 + W_2$ (Figure 4.21).

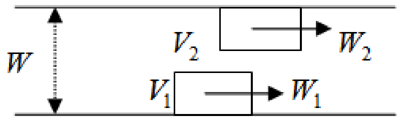


Figure 4.21 Vessels travel in parallel along a link.

4.2.2.4 Multi-link-ahead conflict prediction[3]

The algorithm of conflict prediction is based on an assessment of relative movement of one vessel to the other vessel. Thus, this algorithm is used to predict a conflict risk one link ahead of a vessel's current position. However, it is likely that predicting only one link ahead may not be effective, particularly if the link is short. Therefore, the algorithm is extended to predict a conflict a few links ahead (referred to as multi-link-ahead prediction). With multi-link-ahead prediction, a possible conflict can be predicted a few links ahead. The number of links to look ahead is designed as a parameter in the simulation system. Such a prediction offers the navigator sufficient time to take corrective actions to avoid a conflict before it occurs.

Prediction is performed at current time and estimate the conflict possibility until the meeting time of the two vessels. The time period for conflict prediction is divided into several time intervals. Conflict prediction is to evaluate the relative movement of the domain of one vessel with respect to the domain of another vessel during each time interval. Figure 4.22 shows the revised flowchart of conflict prediction.

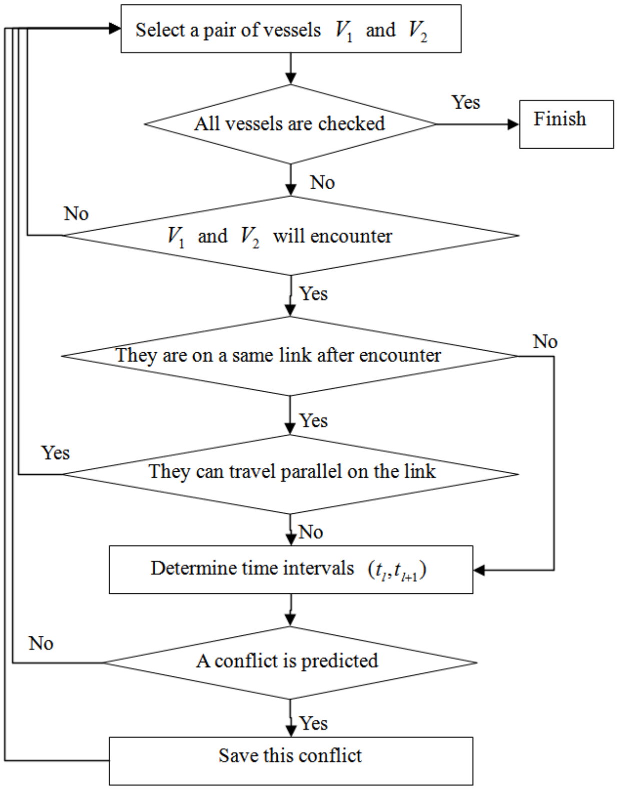


Figure 4.22 Flowchart of conflict prediction.

---

[3] The following publication is based on the results of this section:

Li, Q., Lam J.L.S., and Fan H.S.L. (2012) "Multi-link-ahead Conflicts Prediction in Dynamic Seaport Environments", Serious Games and Simulation.

As shown in Figure 4.23, there are three vessels V_1, V_2 and V_3 whose paths are represented by arrow lines. Potential conflicts may occur in each pair. Paths of V_1 and V_2 intersect at point A which is located near a node. A conflict is likely to occur when the two vessels cross through the node area. Likewise, the paths of V_2 and V_3 intersect at a point B in a link. In this case the two vessels travel along a same link after point B, and may conflict in the link. Our method used to predict a conflict at a node or in a link is basically the same. The only difference is that link width should be taken into account to predict a conflict in a link. If width of a link is sufficient such that two vessels can travel in parallel, a conflict will not occur between them.

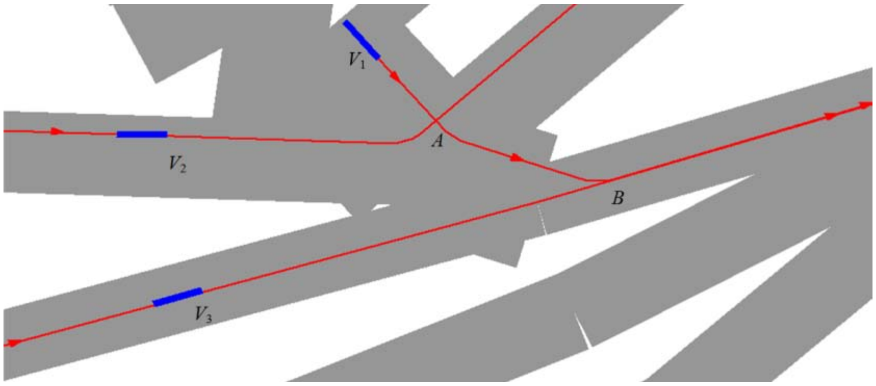


Figure 4.23 Potential conflicts in a traffic network.

An example is used to describe the algorithm for multi-link-ahead conflict prediction (see Figure 4.24). It is assumed that conflict prediction will be executed for several links downstream from the current vessel positions.

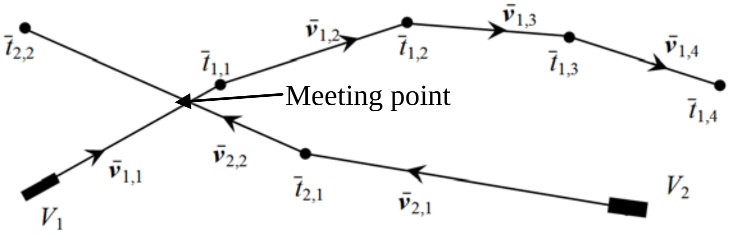


Figure 4.24 Routes of two vessels.

Take P_s as the number of links to look ahead in conflict prediction. The first step in conflict prediction is to estimate whether a pair of vessels will have an encounter on those P_s links. The intersection of the two vessels' routes is defined as the meeting point. Referring to the flowchart in Figure 4.22, if the meeting point exists between these two vessels, the main steps for predicting a potential conflict are as follows:

Step 1: obtain vessel information, including arrival times at the links and speeds along each link.

Step 2: determine time intervals such that both vessels' speeds remain constant in each interval.

Step 3: predict whether there is a conflict in each time interval.

According to Section 4.1.3, the trajectory of a vessel passing P_s links consists of a set of line segmentations, named as sub-links. Each vessel maintains a constant speed and navigation direction in each sub-link. Table 4.4 lists the navigation information for V_1 and V_2 from their current positions to P_s links downstream on their paths. In Table 4.4 and Figure 4.24, the time when V_1 enters the i-th sub-link is $\overline{t}_{1i}$ and V_1 maintains a constant speed $\overline{v}_{1,i}$ on the i-th link. The same applies to V_2. The information in Table 4.4 covers the time intervals $(0, \overline{t}_{1,m})$ and $(0, \overline{t}_{2,n})$ for V_1 and V_2, respectively. In Step 2, to determine whether there is a conflict during the time period $\left(0, \min\{\overline{t}_{1,m}, \overline{t}_{2,n}\}\right)$, the entire time period is divided into a set of time intervals (t_l, t_{l+1}), such that the speeds of V_1 and V_2 are constant (i.e. $v_{1,l}$ and $v_{2,l}$) in each interval. The flowchart in Figure 4.25 shows how the information in Table 4.4 is used to obtain all the time intervals. Step 3 checks, from the first interval to the last interval, whether the two vessels will encounter a conflict. Once a conflict is predicted in a time interval, the algorithm will save the conflict without checking the remaining time intervals.

Table 4.4 Information of two vessels.

Vessels	V_1	V_2
Number of sub-links in the P_s links	m	n
Time to each sub-link	$\overline{t}_{1,i}$, i=1,2,…,m	$\overline{t}_{2,i}$, j=1,2,…n
Velocity on each sub-link	$\overline{v}_{1,i}$, i=1,2,…,m	$\overline{v}_{2,i}$, j=1,2,…n

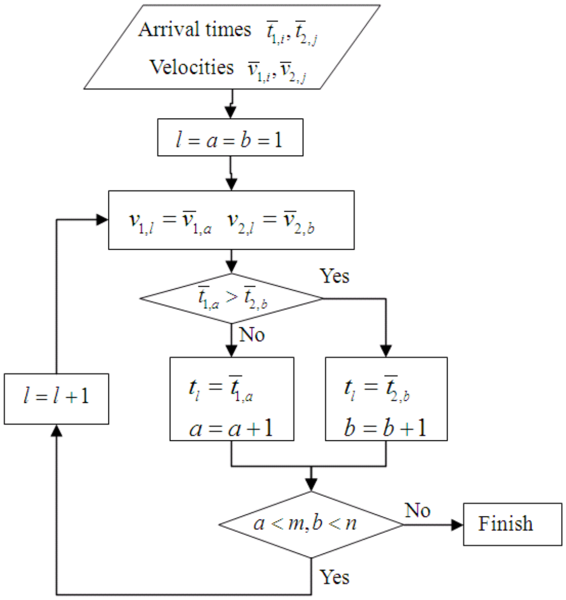


Figure 4.25 Dividing conflict prediction time period into small time intervals.

The criterion to divide the time period into time intervals is to guarantee that the travel direction of each vessel does not change in each time interval. As such, conflict prediction can be done in each time interval. For example,

$$\overline{t_{1,i}} = \{0,3,8,15,20\},$$
$$\overline{t_{2,i}} = \{0,1,7,25\}.$$

Since V_1 does not contain navigation information in the time interval [20,25], thus, the time period for conflict prediction is [0,20]. Combing the two time sequence leads to

$$\overline{t_{1,i}} = \{0,1,3,7,8,15,20,25\}.$$

Thus, the time period is divided into 6 time intervals.

$$[0,1],[1,3],[3,7],[7,8],[8,15],[15,20].$$

In such way, the two vessels remain their traveling direction in each time interval.

In a time interval (t_l, t_{l+1}), the conflict can be predicted using the movement of V_1 relative to V_2, which is the movement of the domain of V_1 with respect to the domain of V_2. Suppose

$w_1 = v_{1,l} - v_{2,l}$: the velocity of V_1 with respect to V_2,

$Q_{i,l} = \left(q_{i,l}^1, q_{i,l}^2, q_{i,l}^3, q_{i,l}^4\right)$: the domain of vessel V_i at $t = t_l$,

$q_{i,l}^k$: the k-th corner of domain $Q_{i,l}$,

$p_{i,l}^{k+1} p_{i,l}^k$: the k-th edge of domain $Q_{i,l}$.

The movement of the corner $q_{1,l}^k$ relative to V_2 is a line segment $p_{1,l}^k q_{1,l}^k$ (Figure 4.26) where

$$p_{1,l}^k = q_{1,l}^k + \left(t_{l+1} - t_l\right) w_{l.}$$

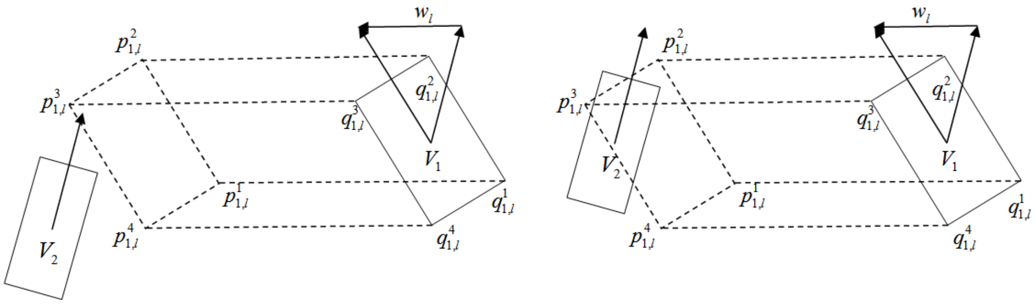
The movement of the edge $q_i^k q_i^{k+1}$ relative to V_2 is a parallelogram $p_i^k = q_{1,l}^k q_{1,l}^{k+1} p_{1.l}^{k+1} p_{1,l}^k$. If V_1 and V_2 conflict with each other, the movement of at least one edge of V_1 will intersect with the domain of V_2, i.e. $P_i^k \cap Q_{2,l} \neq \varnothing$. V_1 and V_2 will conflict in the time interval (t_l, t_{l+1}) if and only if the follow condition is satisfied

$$\bigcup\left(P_i^k \cap Q_{2,l}\right) \neq \varnothing.$$

Conflict prediction is equivalent to check whether a parallelogram intersect with a rectangle. The example in Figure 4.26(a) shows no conflict between V_1 and V_2 in the time interval (t_l, t_{l+1}), because $\bigcup P_i^k \cap Q_{2,l} \neq \varnothing$. However, the situation shown in Figure 4.26 (b) gives

$$P_l^1 \cap Q_{2,l} = \varnothing, P_l^2 \cap Q_{2,l} \neq \varnothing, P_l^3 \cap Q_{2,l} \neq \varnothing, P_l^4 \cap Q_{2,l} = \varnothing.$$

We can see that, there is a conflict between V_1 and V_2. The algorithm will proceed from the first parallelogram P_l^1 to the fourth parallelogram P_l^4, to check if the intersections (i.e. $P^k_i \cap Q_{2,l}$) are empty. If one parallelogram intersects with $Q_{2,l}$, a conflict will occur and will therefore be predicted.



(a) (b)

Figure 4.26 Predicting vessel conflict in time interval (t_l, t_{l+1}).
(a) no conflict between V_1 and V_2, and (b) there will be a conflict between V_1 and V_2.

4.3 Examples and discussions

In this section, some examples are shown to illustrate the application of conflict prediction algorithms in different scenarios. Input data used here, such as vessel parameters, routes and schedules, are not real data but are randomly generated by a simulation system (see Chapter 6 for details).

4.3.1 An example for single conflict prediction

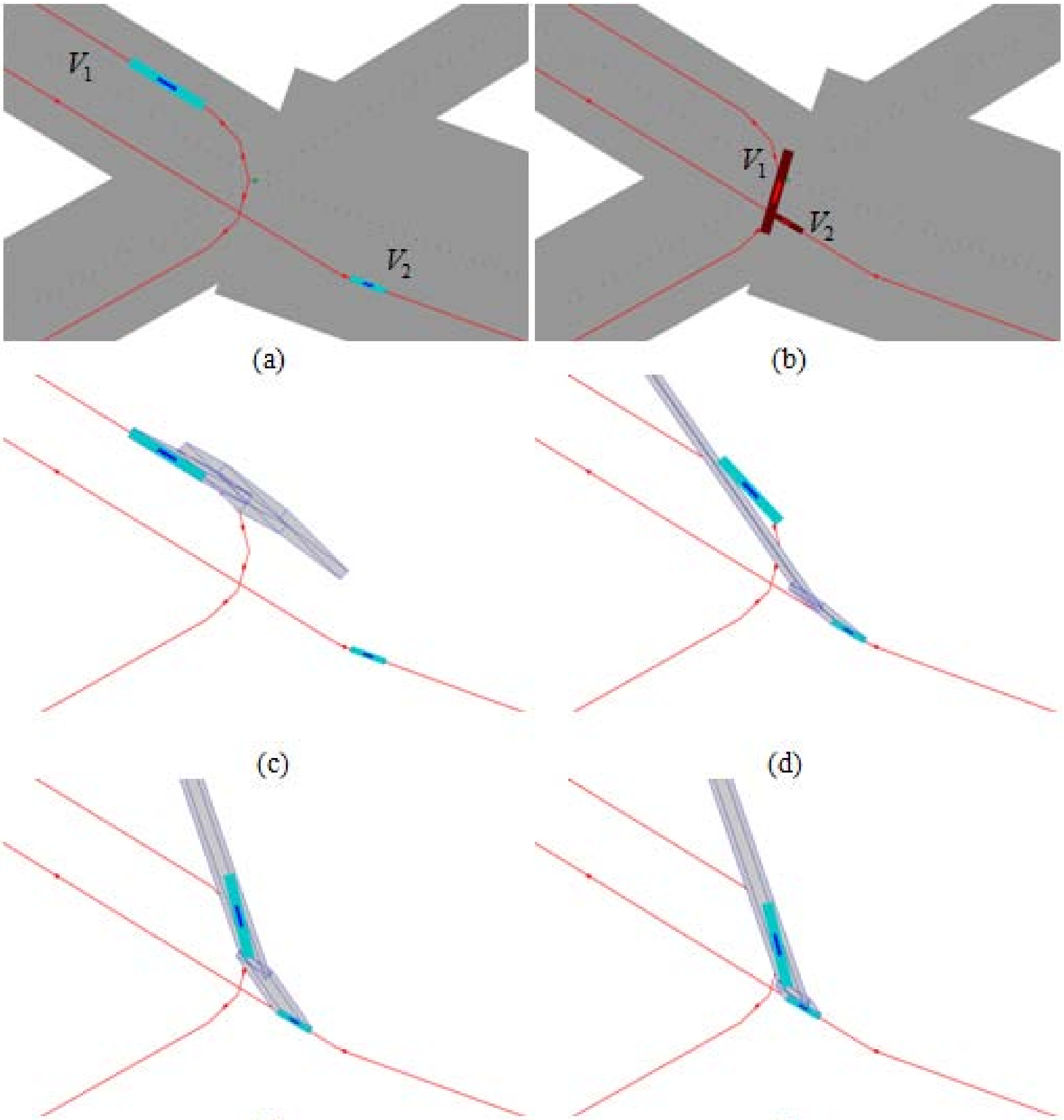
Figure 4.27 is an example of a node conflict between two vessels: V_1 and V_2. Information of the vessels is listed in Table 4.5.

Table 4.5 The two vessels.

Vessels	LOA (m)	Beam (m)	Speed (knots)	Domain length (m)	Domain width (m)
V_1	40	12	10	120	36
V_2	90	18	9	90	18

As shown in Figure 4.27(a), V_1 and V_2 are moving towards the same node. The arrow lines in red color represent the vessel trajectories. If the two vessels keep moving along their given trajectories, their domains will intersect as shown in Figure 4.27(b). The conflict prediction method will be used to illustrate that the conflict can be predicted before the two vessels reach the meeting point. Figure 4.27(c-f) shows the changes of vessel movements from they are far apart until they encounter. The relative movements are represented by the parallelograms enclosed by solid lines. In this example two-link-ahead prediction is used, and the time period for conflict prediction is divided into three intervals. The conflict is accurately predicted at the third time interval in Figure 4.27(d). Likewise, it is predicted at the second time interval in Figure 4.27(e) as they come closer.

In Figure 4.27(c), the gray areas indicate the relative movement between V_1 and V_2 when they are at the positions shown in Figure 4.27(a). It is clear that the relative movement of V_1 does not intersect with the domain of V_2. Thus there is no conflict predicted at this time. As the two vessels move closer to the node, the relative movement of V_2 intersects with the domain of V_1, as shown in Figure 4.27(d-f). Therefore, a conflict is predicted.



(a) (b)

(c) (d)

(e) (f)

Figure 4.27 Two vessels from different links conflict at a node.
(a) V_1 and V_2 moving towards the same node, (b) a conflict between V_1 and V_2, (c) the conflict is not predicted at this position, and (d-f) the conflict is predicted at different positions.

4.3.2 An example of predicting multiple conflicts

The previous example shows that the conflict prediction algorithm developed here can effectively predict a single conflict between two vessels. This algorithm also works effectively with several vessels and multiple potential conflicts. An example of 12 vessels is run in the simulation system. Information of the vessels is listed in Table 4.6.

Table 4.6 The 12 vessels.

Vessels	LOA (m)	Beam (m)	Speed (knot)	Domain length (m)	Domain width (m)
vessel1	330	50	8	990	150
vessel2	180	30	10	540	90
vessel3	90	18	9	270	54
vessel4	270	32	12	810	96
vessel5	270	32	12	810	96
vessel6	110	20	8	330	60
vessel7	110	20	8	330	60
vessel8	330	50	8	990	150
vessel9	240	38	8	720	114
vessel10	270	32	12	810	96
vessel11	180	30	10	540	90
vessel12	110	20	8	330	60

Figure 4.28 is a screen snap shot when there are 10 vessels traveling in a traffic network. Conflicts statistics are collected and summarized in Table 4.7, where

T_d: time when a conflict is detected using the conflict determination method, (the format for a time is hour : minute : second),

T_p: time when a conflict is predicted using the conflict prediction method,

T_s: lead time in seconds that a conflict can be predicted before it actually occurs, i.e. the difference between T_d and T_p,

P_s: the number of links to look ahead in conflict prediction.

In this case, four conflicts are detected using the conflict determination method. The conflict prediction is tested on P_s = 2, 3 and 4. The worst case involves the fourth conflict which is predicted only 48 seconds before it actually occurs with P_s = 2. This may not be enough for a vessel to take corrective action to avoid the conflict. Thus, increasing P_s is considered. Clearly, conflict risks can be predicted earlier when the number of links to look ahead is increased to P_s = 3 or 4.

Theoretically, multi-link-ahead prediction can predict all the conflicts that a vessel will be involved in at the very beginning when the vessel enters the seaport by setting the parameter of P_s as a large value. However, this will increase computation time, which is not desirable. A proper value of P_s will be set in the simulation stage.

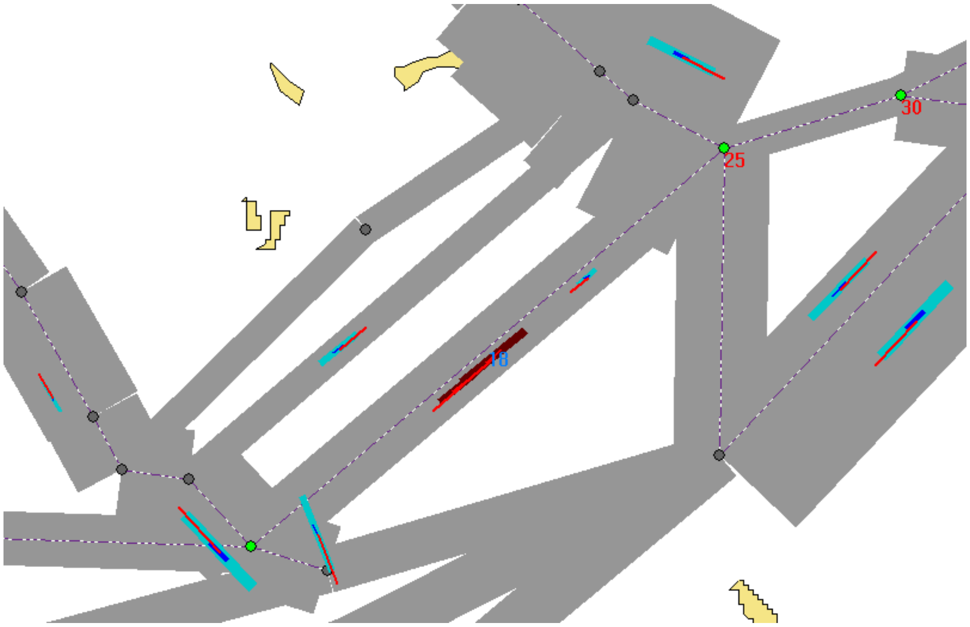


Figure 4.28 A screen snap shot of a detected conflict.

Table 4.7 Conflict statistics.

Vessels	Location	T_d	Conflict prediction					
			$p_s = 2$		$p_s = 3$		$p_s = 4$	
			T_p	T_s	T_p	T_s	T_p	T_s
vessel2 and vessel10	LINK18	8:54:30	8:46:12	498	8:42:48	702	8:42:48	702
vessel3 and vessel10	LINK18	8:48:00	8:46:12	132	8:45:12	168	8:42:42	318
vessel5 and vessel8	NODE30	8:32:06	8:27:42	264	8:26:48	318	8:26:48	318
vessel8 and vessel12	NODE25	8:28:36	8:27:48	48	8:24:18	258	8:23:42	294

4.4 Summary

Algorithms were designed and implemented to determine and predict a conflict using the criterion of ship domain. In the case of two vessels encountering each other, a conflict occurs when a domain of one vessel enters the domain of the other.

Conflict determination was to evaluate the relative positions of the two rectangles concerned, which is implemented in two steps. Firstly, a pre-processing was used to eliminate the cases where vessels are far apart from each other. Then the main algorithm was executed to evaluate the relationship between the two vessels' domains. Conflict determination can be used in the preplanning stage to determine potential conflicts under the original schedules of the vessels.

An algorithm was designed to predict a conflict through the estimation of vessel positions a few links in advance (called here multi-link-ahead). It allows sufficient time for navigators to take evasive actions to achieve conflict avoidance. The algorithm simplifies the conflict prediction problem by checking whether two parallelograms intersect with each other. Conflict prediction algorithm with multi-link look-ahead is a particular contribution of this research, which has never been considered by other studies. Conflict prediction will be used in the simulation stage in conflict resolution so that corrective actions can be taken to prevent the potential conflicts from occurring. Simulation results showed that conflicts can be accurately predicted in time.

Chapter 5

Conflict resolution

Based on the conflict detection algorithms in the previous chapter, a two-stage strategy will be proposed in this chapter to systematically resolve possible conflicts within a seaport. The first stage is preplanning which will detect and resolve potential conflicts in a deterministic manner using the original schedule of vessel arrivals and departures. This then will produce a revised schedule of vessel arrivals and departures which minimize delays and reduce congestion by absorbing the major part of delay outside the port waters. The second stage will take care of the stochastic part due to random fluctuations in vessel speed, deviation from schedules, which may result in potential conflicts in actual vessel movements.

A simulation platform will be presented in the following chapter for implementing such conflict resolution strategy in a dynamic traffic environment.

5.1 Introduction

In this chapter, conflict resolution is implemented in two stages:

- The preplanning stage: given schedules for all vessels, before vessels begin traveling into the seaport, the system will perform the preplanning to provide a new schedule for each vessel so as to remove the potential conflicts.
- The simulation stage: after vessels reach the seaport following the new schedules, our system will simulate vessel movements with randomness and provides corrective measures for vessels to resolve conflicts.

Vessels within the port traffic system are supposed to travel along regular routes according to given schedules. It is based on a general rule that, a vessel before arriving is required to report to the seaport its route, arrival time, and vessel characteristics, for the purpose of reserving a berthing/anchorage area for loading/unloading or ordering services for maintenance, piloting, and refueling. However, potential conflicts and even collision accidents may exist under original schedules which are collected from the arrival reports. To enable safe navigation and high performance of traffic network, a function of preplanning is developed to preprocess the original schedules before the actual simulation starts. The preplanning establishes initial schedule chart for the traffic system, where most of the possible conflicts are eliminated in advance, and then leaves relatively few cases that have random fluctuations which result in potential conflicts for the simulation system to resolve.

After preplanning, each vessel is assigned with a new preplanned schedule. However, there will be fluctuations in the schedule for each vessel. For example, the actually arrival time of each vessel may be slightly different from the pre-assigned schedule, and the actually time for each vessel to leave the seaport may also vary due to the uncertainty in the loading/unloading working time. Thus, there are potential conflicts in real vessel movements. A simulation system will be presented for simulating vessel movements and to resolve possible conflicts by corrective measures.

5.2 The preplanning stage

The preplanning aims to eliminate potential conflicts under original vessel schedules. In this stage, a rescheduling process will be executed to adjust vessels' arrival times so as to coordinate vessel movements without conflicts. The idea of rescheduling is to satisfy the minimum time separation between any pair of successive vessels arriving at links and nodes. A node-based rescheduling algorithm (Fan 1988; Fan and Cao 2000) will be implemented for this purpose. However, an analysis shows that the algorithm has two limitations and thus cannot work properly for the preplanning task. Alternatively, a vessel-based rescheduling algorithm is proposed and implemented here.

5.2.1 Preliminaries

This section presents basic notations which will be used in algorithm development, including the schedule chart, the priority list, the schedule tree, the minimum time separation, and the arrival time table.

5.2.1.1 Schedule chart

Assume that there are *n* vessels in the traffic system

$$V = \left\{ V_1 , V_2 , \dots , V_n \right\}.$$

The initial schedule chart S^0 consists of the originally vessel schedules denoted as

$$S^0 = \left\{ s_1^0 , s_2^0 , \dots , s_n^0 \right\} , \tag{5.1}$$

where s_i^0 is the schedule of the vessel v_i .

If no conflict is detected under S^0, this schedule chart can be referred to as a feasible one. Otherwise, S^0 needs to be adjusted so as to obtain a feasible schedule chart where no conflict occurs between any two vessels. The new feasible schedule can be denoted as

$$\overline{S} = \left\{ \overline{s}_1 , \overline{s}_2 , \dots , \overline{s}_n \right\}. \tag{5.2}$$

The minus of the two schedules provides the delay information for each vessel as

$$\Delta = S^0 - \overline{S} = \left\{ \delta_1 , \delta_2 , \dots , \delta_n \right\} , \delta_i \geq 0, \tag{5.3}$$

where δ_i is a time period that the vessel v_i is postponed.

On the other hand, applying a delay Δ onto a schedule chart $\overline{S}$ produces a new schedule chart $\hat{S}$ such that $\overline{S} - \hat{S} = \Delta$.

5.2.1.2 Event-driven conflict determination and time-driven conflict determination

In the preplanning stage, vessels are supposed to be rescheduled iteratively until there is no conflict between ant two vessels in the operation of the traffic system. The rescheduling process is to execute a loop between conflict detection and schedule adjustment on the condition that: as long as conflicts are detected, adjustment for vessel schedules will be made. As discussed in Chapter 4, a conflict can be determined through evaluating the event of domain interference. Such method is referred to as event-driven conflict determination. However, it is complicated to estimate the relative position between two vessels, considering the dimensions of vessels and their domains. In the preplanning stage, the event-driven method is very time-consuming to detect all conflicts under the given schedule chart, especially for a huge number of vessels.

A vessel travelling in the port needs to keep adequate time separation to safely leave behind the vessel in front of it, which can be referred to the minimum time separation requirement. In the preplanning stage, it is practical to apply the criterion of the minimum time separation for conflict determination, which is called as time-driven method. With the time-driven method, a conflict can be easily determined through comparing the time interval of two successive vessel arrivals with the minimum time separation required between them (details are shown in Section 5.2.1.3). The time-driven conflict determination enables fast calculation in the rescheduling process and thus is more practical than the event-driven method.

It is admitted that, the event-driven conflict determination is more accurate than the time-driven method. However, accuracy of conflict determination is not a necessary consideration in the preplanning stage, because vessel movements are random due to fluctuations in vessel arrivals in real-time. The preplanning does not aim to eliminate all conflicts before the actual vessel arrivals, but it provides a meaningful schedule chart such that the number of potential conflicts under original vessel schedules will be highly reduced. This then leaves relatively few cases that have random fluctuations which result in potential conflicts for the simulation stage to resolve.

On the contrary, the event-driven conflict determination will be used as a tool for evaluating the results of conflict resolution, for instance, whether conflicts are effectively resolved (in Section 5.2.5, Section 5.3.6 and Section 6.3.4). In such cases, only a small amount of conflicts will be involved. Thus the event-driven method can accurately detect such conflicts with efficiency.

5.2.1.3 The minimum time separation

The rescheduling is to guarantee the minimum time separation between any two successive vessels traveling in the traffic network. The minimum time separation between two vessels depends on the minimum distance separation (inclusive of the domains of the vessels) and relative speed of the vessels.

The minimum time separation in a link

Suppose two vessels V_1 and V_2 are travelling along a link L at speeds v_1 and v_2. Referring to Figure 4.1, $\overline{L}^1$ and $\overline{L}^2$ are the lengths of clearance distance of the two vessels, respectively (Figure 5.1).

It is obvious that, no conflict will occur between the two vessels if $v_1 < v_2$. If a potential conflict occurs between the two vessels, this implies that $v_1 \geq v_2$. A conflict occurs in the link, as long as V_1 catches up with V_2 or they are too close. Figure 5.2 shows a threshold of conflict occurrence when V_1 falls behind with V_2 at the end of a link.

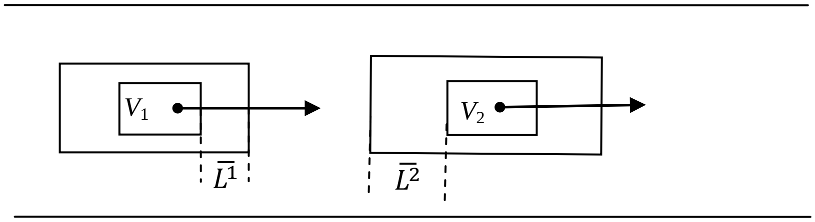


Figure 5.1 Vessels travelling along a link.

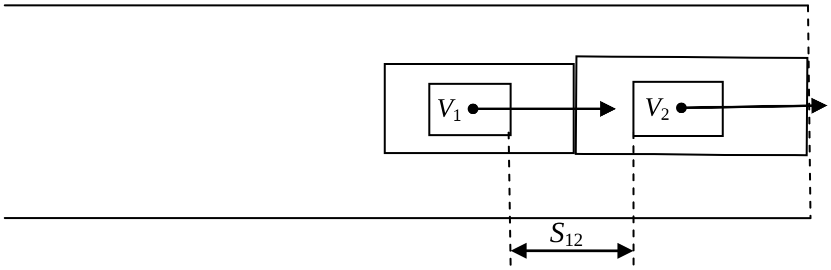


Figure 5.2 The distance separation at the end of a link.

Thus, the required minimum distance separation S_{12} between the two vessels is

$$S_{12} = \overline{L}^1 + \overline{L}^2 . \tag{5.4}$$

And, the minimum time separation T_{12}^{link} between the two vessels is

$$T_{12}^{link} = S_{12} / (v_1 - v_2), \quad v_1 > v_2 . \tag{5.5}$$

The minimum time separation at a node

Suppose two vessels of V_1 and V_2 are crossing a node at speed v_1 and v_2, respectively. Vessel trajectory of V_1 is from the link L_{11} to the link L_{12}, and vessel trajectory of V_2 is from the link L_{21} to the link L_{22}, as shown in Figure 5.3.

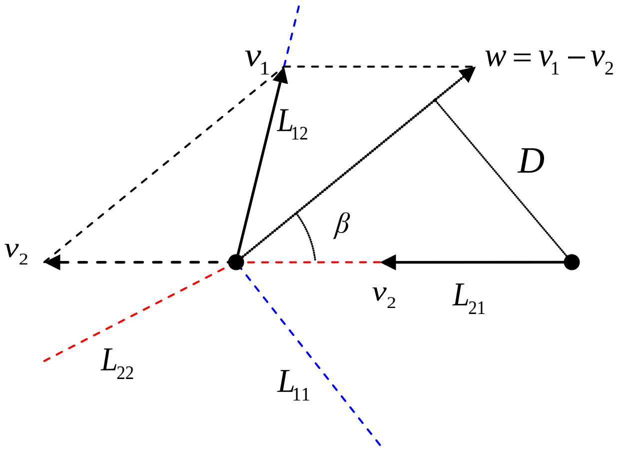


Figure 5.3 Separation between two vessels crossing a node.

Suppose that the crossing starts at time t_1, and V_1 arrives at the node while V_2 is approaching the node. In order to ensure the minimum distance separation to V_1, V_2 is required to arrive at the node at $\overline{t_2}$, which is called the safe arrival time. Thus, at time t_1, the distance S between V_2 and the node is

$$S = \|v_2\| \left(\overline{t_2} - t_1 \right).$$

The speed of V_1 relative to V_2 is $w = v_1 - v_2$. Suppose the angle between w and $-v_2$ is β as shown in Figure 5.3. The minimum distance separation D between the two vessels should be

$$D = S \cdot \sin\beta.$$

To avoid a conflict, it should be guaranteed that

$$\begin{aligned} D \geq S_{12} \quad & \Rightarrow S \cdot \sin\beta \geq S_{12}, \\ & \Rightarrow \|v_2\| \left(\overline{t_2} - t_1 \right) \geq \frac{S_{12}}{\sin\beta}, \\ & \Rightarrow \overline{t_2} \geq \frac{S_{12}}{\|v_2\| \cdot \sin\beta} + t_1. \end{aligned}$$

where S_{12} is the minimum distance separation between V_1 and V_2, as defined in Eq.(5.4).

The minimum safe arrival time $\overline{t_2}$ for V_2 should be

$$\overline{t_2} = \frac{S_{12}}{\|v_2\| \cdot \sin\beta} + t_1.$$

Therefore, when V_1 crosses the node, the required minimum time separation T_{12}^{cross} between the two vessels is

$$T_{12}^{cross} = \overline{t_2} - t_1 = \frac{S_{12}}{\|v_2\| \cdot \sin\beta}. \tag{5.6}$$

There is another way to determine T_{12}^{cross}. Suppose that, when V_1 crosses the node, the angle between v_1 and $-v_2$ is α (in Figure 5.4).

The relation between the angle α and the angle β satisfies,

$$\sin\beta = \frac{A}{B} = \frac{D}{C} = \frac{\|v_1\| \sin\alpha}{\sqrt{\|v_1\|^2 + \|v_2\|^2 + 2\|v_1\| \cdot \|v_2\| \cdot \cos\alpha}}.$$

The safe arrival time $\overline{t_2}$ for V_2 is

$$\overline{t_2} = \frac{S_{12}}{\|v_2\| \cdot \sin\beta} + t_1 = \frac{S_{12}\sqrt{\|v_1\|^2 + \|v_2\|^2 + 2\|v_1\| \cdot \|v_2\| \cdot \cos\alpha}}{\|v_1\| \cdot \|v_2\| \sin\alpha} + t_1.$$

Therefore, the required minimum time separation T_{12}^{cross} between the two vessels is

$$T_{12}^{cross} = \overline{t_2} - t_1 = \frac{S_{12}\sqrt{\|v_1\|^2 + \|v_2\|^2 + 2\|v_1\| \cdot \|v_2\| \cdot \cos\alpha}}{\|v_1\| \cdot \|v_2\| \sin\alpha}. \tag{5.7}$$

Between the two crossing vessels, the minimum time separation depends on the angle of their directions. In Eq.(5.7), when α increases in the range of $(0, \pi/2]$, T_{12}^{cross} will decrease accordingly.

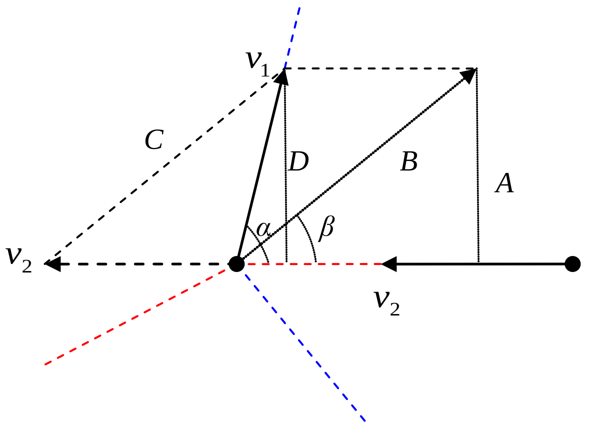


Figure 5.4 Two vessels crossing a node.

5.2.1.4 Priority list of vessels

Adjustments of original schedule chart will result in delays for certain amount of vessels arriving in the seaport. Vessel priority is taken into account in the preplanning stage. It means that, in the rescheduling process, necessary delay would be added into the schedule of the vessel whose priority is lower. The priority value for each vessel should be assigned by the traffic control center. Each vessel can also bid for its priority before its arrival. A higher priority will lead to less delay.

Suppose that the priority values of all vessels are listed and sorted in ascent order, which is denoted as

$$\Psi = \{l_1, l_2, \cdots, l_n\}, \tag{5.8}$$

where $l_i \in \{1, 2, \cdots, n\}$.

Different priority lists will lead to different solutions for the rescheduling. For *n* vessels, the number of different priority lists can be up to $n! = n(n-1)(n-2)\cdots 1$.

5.2.1.5 Schedule tree

Different priority lists may introduce different delays in vessels, thus lead to different simulation results. The number of priority lists can be huge. Listing all the results becomes impossible. The schedule tree is used here aiming to organize different schedule results.

The traditional schedule tree was used to find the best order of processing *n* jobs on *m* machines such that the total processing time for all jobs is minimized (Lomnicki 1965). It can be modified to solve different optimization problems. For example, the schedule tree can be used in a seaport to solve the yard crane dispatching problem (Guo *et al.* 2009). Usually, a schedule tree consists of tree nodes, each of which corresponds to a schedule. In this section, a new structure of the schedule tree is proposed aiming to obtain an optimized schedule chart. Each tree node in the proposed schedule tree represents a schedule charts. Thus, a schedule tree is a tree structure of schedule charts. The schedule tree can be constructed using the following rules:

- The tree root N_1 is the initial schedule chart S^0 (Figure 5.5).
- Each tree node is a schedule chart. If the schedule chart has conflicts, it may have children nodes. Otherwise, it is a leaf node without any child. For example, $N_{1,2}$ and $N_{1,1,2,3}$ in Figure 5.5 are leaf nodes.
- For a non-leaf node N_i, applying one delay $\Delta_{i,j}$ introduces a child $N_{i,j}$ for N_i. In Figure 5.5, applying $\Delta_{1,3}$ on N_1 introduces a node $N_{1,3}$, and applying $\Delta_{1,3,1}$ on $N_{1,3}$ creates $N_{1,3,1}$.

According to the construction of the schedule tree,

$$N_i - N_{i,j,k} = \left(N_i - N_{i,j}\right) - \left(N_{i,j} - N_{i,j,k}\right) = \Delta_{i,j} + \Delta_{i,j,k}. \tag{5.9}$$

Appling the delay $\Gamma_{i,j,k} = \Delta_{i,j} + \Delta_{i,j,k}$ onto the node N_i leads to the node $N_{i,j,k}$. Therefore, Figure 5.5 can be simplified to Figure 5.6, with

$$\begin{aligned}
&\Gamma_{1,1,1} = \Delta_{1,1} + \Delta_{1,1,1}, \\
&\Gamma_{1,1,2,1} = \Delta_{1,1} + \Delta_{1,1,2} + \Delta_{1,1,2,1}, \\
&\Gamma_{1,1,2,2} = \Delta_{1,1} + \Delta_{1,1,2} + \Delta_{1,1,2,2}, \\
&\Gamma_{1,1,2,3} = \Delta_{1,1} + \Delta_{1,1,2} + \Delta_{1,1,2,3}, \\
&\Gamma_{1,2} = \Delta_{1,2}, \\
&\Gamma_{1,3,1} = \Delta_{1,3} + \Delta_{1,3,1}.
\end{aligned}$$

Two tree nodes N_i and N_j are equivalent if and only if the two delays are the same

$$N_i - S^0 = N_j - S^0. \tag{5.10}$$

In Figure 5.5, if $\Gamma_{1,1,1} = \Delta_{1,1} + \Delta_{1,1,1} = \Delta_{1,3} + \Delta_{1,3,1} = \Gamma_{1,3,1}$, then the two tree nodes $N_{1,1,1} = N_{1,3,1}$.

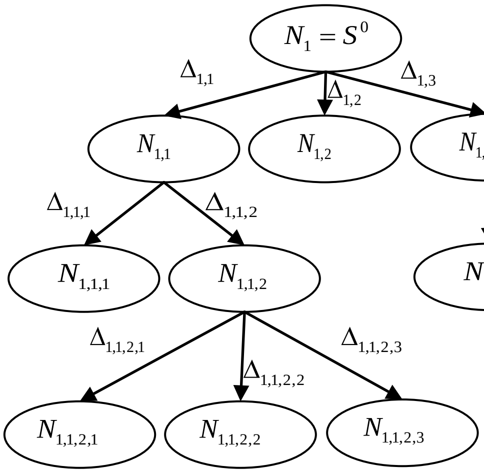


Figure 5.5 A schedule tree.

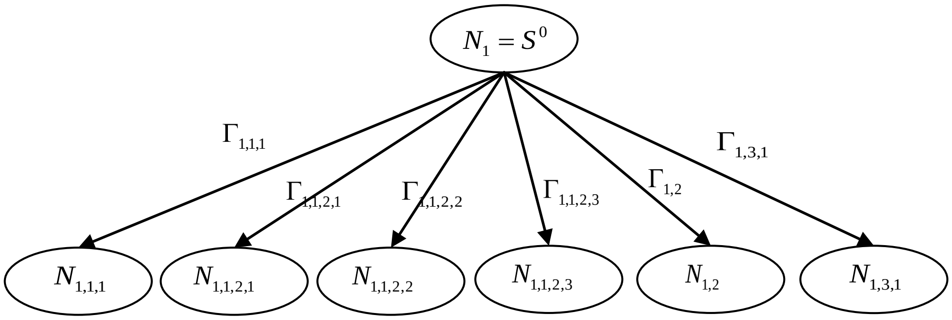


Figure 5.6 Another tree equivalent to Figure 5.5.

5.2.1.6 Arrival time table on a node

From a given schedule chart, the time that a vessel reaches each node can be estimated. An arrival time table for a node in the traffic network collects all the traveling information of the vessels passing the node. Table 5.1 is an example of the arrival time table on a node , where

V_x : ID of the vessel passing through the node,
t_x^y : time of V_x arriving at the node, ranked the y^{th} in order of arrival,
L_x : the link that V_x comes from,
$L_x^{'}$: the link that V_x will move to.

In the arrival time table, vessels are sorted according to their arrival time in ascend order.

Table 5.1 Arrival time table on a node.

Vessel ID	Arrival time	From the link	To the link
V_i	t_i^1	L_i	$L_i^{'}$
V_j	t_j^2	L_j	$L_j^{'}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
V_x	t_x^y	L_x	$L_x^{'}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

5.2.2 Method description

In preplanning stage, a rescheduling process is used to adjust time interval between vessel arrivals according to criterion of the minimum time separation.. To avoid conflicts and traffic congestion, the port traffic controllers in planning of schedules would require an arrival rate of given vessels to be consistent with acceptable rate of fairways. A vessel travelling in fairways needs to keep adequate time separation to be safely behind the vessel in front of it, which can be referred to the minimum time separation requirement. The simulation system proposed in this research provides a platform for the traffic controllers to execute the preplanning so as to avoid conflicts and congestion ahead of time, i.e. next day.

A preplanning method that is developed by Fan *et al* (Fan 1988; Fan and Cao 2000) performs the rescheduling process node-by-node. A counter-example will be presented to show that the node-based algorithm may not always guarantee a solution for the conflict-free rescheduling. To overcome the drawbacks, a new preplanning method will be proposed to perform the rescheduling process vessel-by-vessel. Meanwhile, appropriate optimization techniques are applied in the rescheduling process for the purpose of delay minimization with use of a schedule tree.

5.2.3 Node-based rescheduling

5.2.3.1 Algorithm

The idea of the rescheduling algorithm is to adjust time interval between any a pair of successive vessels arriving at nodes so as to satisfy the minimum time separation between them. Nodes in the traffic network are assigned as recording points. Times of vessel arrivals at nodes will be estimated under the original schedules. An arrival time table is established at each node, which records arrival times of all vessels passing the node. In the arrival list, threshold times for each pair of consecutive arrivals are compared with the required time separation between them. If the separation is inadequate, the vessel behind will be delayed so as to extend the time interval to satisfy the minimum separation requirement. Noted that the node-based preplanning is based on first-come-first-serve basis, i.e. priority of the vessel is set according to its arrival time. The earlier it arrives, the higher priority it has.

Table 5.1 gives an example of the arrival time table of vessels at a certain node. Combining the arrival time table at all node, we could list all arrival times of vessels in the traffic network and sort them in ascend order. The initial arrival time table for all vessels and all nodes is shown in Table 5.2, where t_{xY}^{z} is the time of the vessel V_x arrives at node Y, which is listed at z^{th} in the arrival time table.

Table 5.2 Arrival time table for all vessels and all nodes.

Arrival time	Vessel ID	Location (Node ID)
t_{iJ}^{1}	V_i	J
t_{mN}^{2}	V_m	N
…	…	…
t_{xY}^{z}	V_x	Y
…	…	…

In the arrival timetable, time interval of each pair of successive arrivals on a same node will be compared with the minimum time separation required between them. A potential conflict may occur when the required separation is violated. The idea of eliminating the conflict is to adjust the vessel behind in the conflict situation (i.e. later arrival in arrival time table) with a certain amount of delay, thus extending time interval to satisfy the separation requirement. For two vessels V_i and V_j passing a same node, suppose

S_{ij} : the minimum time separation required between V_i and V_j.

T_{ij} : actual time separation between V_i and V_j.

The necessary delay d to guarantee the minimum time separation can be calculated by

$$d = S_{ij} - T_{ij}. \tag{5.11}$$

Figure 5.7 is the flowchart of the node-based preplanning. Algorithm of a preplanning process is described as follows:

- Step 1: Establish original arrival time table.
- Step 2: Initialize $k = 1$.
- Step 3: Select t_{aB}^{k} in arrival time table.

  - Find the first t_{cD}^{l} after the (k+1)-row in the table such that t_{aB}^{k} and t_{cD}^{l} are arrival times at a same node (i.e. B=D). If no t_{cD}^{l} found, go to Step 4.
  - Check whether (t_{cD}^{l} - t_{aB}^{k}) satisfies the required time separation between them. If yes, go to Step 4.
  - Add necessary delay d_{cD} to the vessel V_c such that the minimum time separation is guaranteed.
  - Update and sort the arrival time table. Then go to Step 2.
- Step 4: If t_{aB}^{k} is listed at last in the arrival time table, end. Else, execute ($k = k$ +1), and then go to Step 3.

We can see that, this algorithm is executed based on arrival time tables at each node in the traffic network. Thus, it is called node-based rescheduling algorithm.

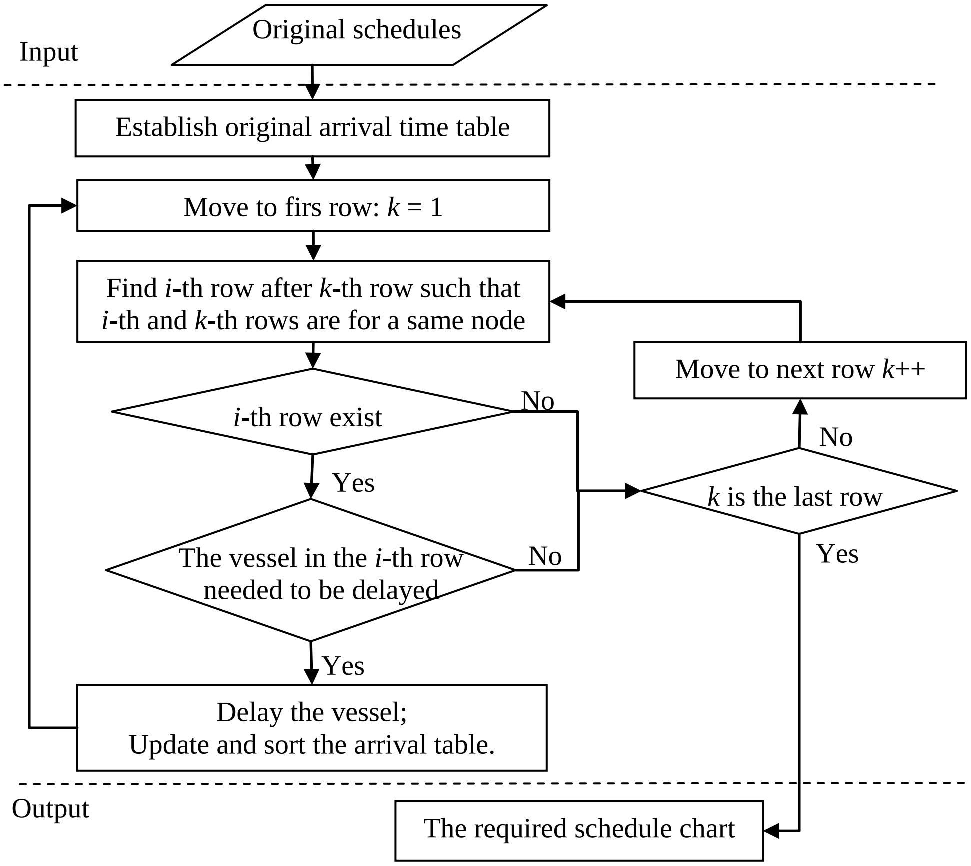


Figure 5.7 The flowchart of node-based preplanning.

5.2.3.2 Limitations and improvements

Node-based rescheduling algorithm has been implemented for preplanning of conflict-free traffic within an airport (Fan 1988; Fan and Cao 2000). However, the algorithm has two limitations when applied for seaport preplanning:

- It cannot eliminate conflicts in links.
- It may generate infinite loops and thus fails to provide a solution.

Link conflicts

An example is provided here to illustrate why the node-based algorithm cannot handle link conflicts. Suppose that in the traffic network, node *A* connects to three links *BA*, *CA* and *DB* (Figure 5.8) and each link can host only one vessel. An arrival time table is established at node *A*, as shown in Table 5.3. The three vessels V_a, V_b, and V_c will arrive at *A* at arrival times t_a, t_b, and t_c, respectively.

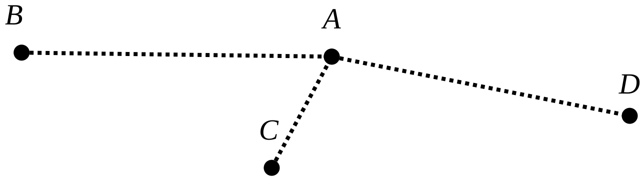


Figure 5.8 A part of the traffic network.

Table 5.3 Arrival time table at node *A*.

Vessel ID	Arrival time	From the link	To the link
V_a	t_a	*BA*	*DA*
V_b	t_b	*BA*	*CA*
V_c	t_c	*CA*	*DA*

For two consecutive arriving vessels, if they come from same link before arriving at *A*, a conflict in a link may occur; otherwise, there may be a conflict in an intersection (i.e. node *A*). From the information in Table 5.3, each pair of arrivals is from different links, except for the arrivals of V_a and V_b (both are from link BA). Table 5.4 lists each pair of arrivals, time interval between arrivals, and the minimum separation required. It is clear that, a conflict may occur between V_b and V_c, if $\left(t_c - t_b\right) < T_{bc}^{cross}$; Or, a conflict may occur between V_a and V_c, if $\left(t_c - t_a\right) < T_{ac}^{cross}$.

Table 5.4 Information for each pair of arrivals.

A pair of arrivals	Time interval	The minimum time separation
$V_a \& V_b$	$t_b - t_a$	T_{ab}^{cross}
$V_b \& V_c$	$t_c - t_b$	T_{bc}^{cross}
$V_a \& V_c$	$t_c - t_a$	T_{ac}^{cross}

However, to predict a conflict in link BA, one cannot make a simple comparison between the time interval of arrivals $\left(t_b - t_a\right)$ with the minimum separation. Even through the time interval satisfies the minimum separation requirement, there still may be a conflict in the link. Suppose $T_{ab}^{cross} = T_{ba}^{cross}$ is 10 minutes for both node *A* and *B*. As shown in Figure 5.9, V_a arrives at *A* at 1:50pm, and V_b arrives at *A* at 2:10pm. It is obvious that the time interval of 20 minutes is larger than T_{ab}^{cross} (10 minutes). Take the arrival time table at the node *B* into account. Obviously, V_b arrives at node *B* before V_a arriving, but V_a catches up to V_b in the link AB, and thus arrives at *A* earlier than V_b. Thus, there is a conflict in link BA, which is an over-taking situation. This conflict cannot be handled by the original node-based algorithm.

V_b	T_b =1:00pm
V_a	T_a =1:20pm

V_a	t_a = 1:50pm
V_b	t_b =2:10pm

Figure 5.9 Arrival time tables at the *A* and the node *B*.

Correction of link conflicts

The link conflicts discussed above can be corrected. Suppose two vessels V_a and V_b travel on same links from the node *B* to the node *A*, and

T_a : the time when V_a reaches the node *B*,

t_a : the time when V_a reaches the node *A*,

T_b : the time when V_b reaches the node *B*,

t_b : the time when V_b reaches the node *A*.

In general, suppose V_b reaches these links before V_a, i.e. $T_a > T_b$. To avoid potential conflict within these links, V_a should follow behind V_b and keep enough time separation with V_b at the node *A*. Thus, the expected arrival time for V_a to reach the node *A* is

$$t_b + T_{ba}^{cross},$$

where T_{ba}^{cross} is the minimum time separation for V_a following V_b to pass the node *A*.

Then, a potential link conflict may occur between V_a and V_b, when

$$t_a \le t_b + T_{ba}^{cross}. \tag{5.12}$$

In such case, since V_b reached the node *B* before V_a, V_a can be delayed such that it will not catch up with V_b when traveling within these links to avoid the link conflict. The delay value for V_a can be

$$d = t_b - t_a + T_{ba}^{cross}. \tag{5.13}$$

To guarantee the minimum time separation, Eq. (5.11) can be used to calculate the delay for node conflicts and Eq.(5.13) can be used to calculate the necessary delay for link conflicts. For the example in Figure 5.9,

$$d = t_b - t_a + T_{ba}^{cross} = 30\,\text{minutes}.$$

The result is the new time tables in Figure 5.10, as a correction of the time table in Figure 5.9.

V_b	T_b =1:00pm
V_a	T_a =1:50pm

V_b	t_b =2:10pm
V_a	t_a = 2:20pm

Figure 5.10 Arrival time tables after delaying V_a in Figure 5.9.

Infinite loops

The node-based algorithm may fall into infinite loops thus fail to provide a solution. Another counter-example will be presented here.

A simple traffic network is consisting of nodes *A*, *B* and *C*, as shown in Figure 5.11. There are three vessels V_1, V_2, and V_3 within the network. Routes of the vessels are marked with different colors (i.e. V_1: A ➔ C, V_2: A ➔ B, V_3: and B ➔ C.). Following the algorithm, the original arrival time table is shown in Table 5.5. Assume that,

- At node *A*, the time interval of T_{1A} and T_{2A} is less than the separation requirement (i.e. a conflict occurs between V_1 and V_2.). V_1 needs to be delayed for d_{11};
- At node *B*, the time interval of T_{2B} and T_{3B} is less than the separation requirement (i.e. a conflict occurs between V_2 and V_3.). V_2 needs to be delayed for d_{21};
- At node *C*, the time interval of T_{1C} and T_{3C} is less than the separation requirement (i.e. a conflict occurs between V_3 and V_1.). V_3 needs to be delayed for d_{31};

And,

$$d_{31} + d_{21} + T_{2A} < T_{1A}, \tag{5.14}$$

$$d_{11} + d_{21} + T_{1A} < T_{3B}, \tag{5.15}$$

$$d_{31} + d_{11} + T_{3B} < T_{2B}, \tag{5.16}$$

$$d_{11} + d_{31} + T_{2B} < T_{1C}, \tag{5.17}$$

$$d_{11} + d_{21} + T_{1C} < T_{3C}, \tag{5.18}$$

$$d_{21} < d_{11}. \tag{5.19}$$

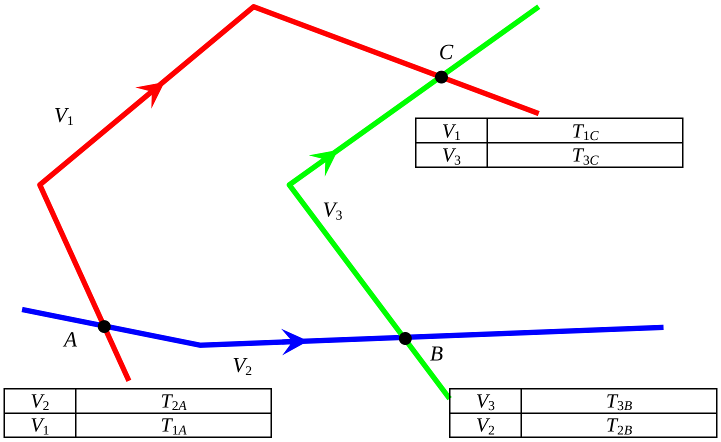


Figure 5.11 A simple network for infinite loops.

Table 5.5 The original time table for the example in Figure 5.11.

Arrival time	Vessel ID	Node ID
T_{2A}	V_2	A
T_{1A}	V_1	A
T_{3B}	V_3	B
T_{2B}	V_2	B
T_{1C}	V_1	C
T_{3C}	V_3	C

The preplanning process is described as follows,

Step 1: T_{2A} and T_{1A} are selected.

- T_{1A} is delayed for d_{11}.
- Add d_{11} to all arrival times of V_1. According to Eq.(5.18), $d_{11}+T_{1C} < T_{3C}$. Update the arrival time table to Table 5.6.

Table 5.6 The time table after the first delay.

Arrival time	Vessel ID	Node ID
T_{2A}	V_2	A
$d_{11}+T_{1A}$	V_1	A
T_{3B}	V_3	B
T_{2B}	V_2	B
$d_{11}+T_{1C}$	V_1	C
T_{3C}	V_3	C

Step 2: T_{2A} and ($d_{11}+T_{1A}$) are selected. No conflict.

Step 3: T_{3B} and T_{2B} are selected.

- T_{2B} is delayed for d_{21}.

- Add d_{21} to all arrival times of $V_{2.}$ According to Eq.(5.19), $d_{21} < d_{11}$. Update arrival time table to Table 5.7.

Table 5.7 The time table after the second delay.

Arrival time	Vessel ID	Node ID
$d_{21}+T_{2A}$	V_2	A
$d_{11}+T_{1A}$	V_1	A
T_{3B}	V_3	B
$d_{21}+ T_{2B}$	V_2	B
$d_{11}+T_{1C}$	V_1	C
T_{3C}	V_3	C

Step 4: ($d_{21}+T_{2A}$) and ($d_{11}+T_{1A}$) are selected.

- ($d_{11}+T_{1A}$) is delayed for d_{21}.
- Add d_{21} to all arrival times of V_1. According to Eq.(5.18), update the arrival time table to Table 5.8.

Table 5.8 The time table after the third delay.

Arrival time	Vessel ID	Node ID
$d_{21}+T_{2A}$	V_2	A
$d_{11}+ d_{21}+T_{1A}$	V_1	A
T_{3B}	V_3	B
$d_{21}+T_{2B}$	V_2	B
$d_{21}+d_{11}+T_{1C}$	V_1	C
T_{3C}	V_3	C

Step 5: ($d_{21}+T_{2A}$) and ($d_{11}+ d_{21}+T_{1A}$) are selected. No conflict.

Step 6: T_{3B} and ($d_{21}+T_{2B}$) are selected. No conflict.

Step 7: ($d_{21}+d_{11}+T_{1C}$) and T_{3C} are selected.

- T_{3C} is delayed for ($d_{31}+d_{21}+d_{11}$).
- Add ($d_{21}+d_{11}+d_{31}$) to arrival times of $V_{3.}$ According to Eq.(5.15), update the arrival time table to Table 5.9.

Step 8: ($d_{21}+T_{2A}$) and ($d_{11}+ d_{21}+T_{1A}$) are selected. No conflict.

Table 5.9 The time table after the forth delay.

Arrival time	Vessel ID	Node ID
$d_{21}+T_{2A}$	V_2	A
$d_{21}+ d_{11}+T_{1A}$	V_1	A
$d_{31}+d_{21}+d_{11}+T_{3B}$	V_3	B
$d_{21}+T_{2B}$	V_2	B
$d_{21}+d_{11}+T_{1C}$	V_1	C
$d_{31}+d_{21}+d_{11}+ T_{3C}$	V_3	C

Step 9: ($d_{31}+d_{21}+d_{11}+T_{3B}$) and ($d_{21}+T_{2B}$) are selected.

- ($d_{21}+T_{2B}$) is delayed for ($d_{31}+d_{21}+d_{11}$).
- Add ($d_{31}+d_{21}+d_{11}$) to arrival times of $V_{2.}$ According to Eq.(5.14), update the arrival time table to Table 5.10.

Table 5.10 The time table after the fifth delay.

Arrival time	Vessel ID	Node ID
$d_{31}+2d_{21}+d_{11}+T_{2A}$	V_2	A
$d_{21}+ d_{11}+T_{1A}$	V_1	A
$d_{31}+d_{21}+d_{11}+T_{3B}$	V_3	B
$d_{31}+2d_{21}+d_{11}+T_{2B}$	V_2	B
$d_{21}+d_{11}+T_{1C}$	V_1	C
$d_{31}+d_{21}+d_{11}+T_{3C}$	V_3	C

Step 10: ($d_{31}+2d_{21}+d_{11}+T_{2A}$) and ($d_{21}+ d_{11}+T_{1A}$) are selected.

- ($d_{21}+ d_{11}+T_{1A}$) is delayed for ($d_{31}+d_{21}+d_{11}$).
- Add ($d_{21}+d_{11}+d_{31}$) to all arrival times of V_1. According to Eq.(5.18), update the arrival time table to Table 5.11.

Table 5.11 The time table after the sixth delay.

Arrival time	Vessel ID	Node ID
$d_{31}+2d_{21}+d_{11}+T_{2A}$	V_2	A
$d_{31}+2d_{21}+2d_{11}+T_{1A}$	V_1	A
$d_{31}+d_{21}+d_{11}+T_{3B}$	V_3	B
$d_{31}+2d_{21}+d_{11}+T_{2B}$	V_2	B
$d_{31}+2d_{21}+2d_{11}+T_{1C}$	V_1	C
$d_{31}+d_{21}+d_{11}+T_{3C}$	V_3	C

Step 11: ($d_{31}+2d_{21}+d_{11}+T_{2A}$) and ($d_{31}+2d_{21}+2d_{11}+T_{1A}$) are selected. No conflict.

Step 12: ($d_{31}+d_{21}+d_{11}+T_{3B}$) and ($d_{31}+2d_{21}+d_{11}+T_{2B}$) are selected. No conflict.

Step 13: ($d_{31}+2d_{21}+2d_{11}+T_{1C}$) and ($d_{31}+d_{21}+d_{11}+T_{3C}$) are selected.

- ($d_{31}+d_{21}+d_{11}+T_{3C}$) are delayed for ($d_{31}+d_{21}+d_{11}$).
- Add ($d_{21}+d_{11}+d_{31}$) to all arrival times of V_3. According to Eq.(5.15), update the arrival time table to Table 5.12.

Table 5.12 The time table after the seventh delay.

Arrival time	Vessel ID	Node ID
$d_{31}+2d_{21}+d_{11}+T_{2A}$	V_2	A
$d_{31}+2d_{21}+2d_{11}+T_{1A}$	V_1	A
$2d_{31}+2d_{21}+2d_{11}+T_{3B}$	V_3	B
$d_{31}+2d_{21}+d_{11}+T_{2B}$	V_2	B
$d_{31}+2d_{21}+2d_{11}+T_{1C}$	V_1	C
$2d_{31}+2d_{21}+2d_{11}+T_{3C}$	V_3	C

From the above steps, it can be summarized that, in the preplanning process, the three vessels will be delayed repeatedly following the sequence: $V_1 \Rightarrow V_3 \Rightarrow V_2 \Rightarrow V_1 \Rightarrow V_3 \Rightarrow V_2 \Rightarrow V_1 \Rightarrow V_3 \Rightarrow V_2 \Rightarrow$…. Thus, an infinite loop occurs.

Resolution of infinite loops

Consider the preplanning process as a mathematical problem. There are three vessels V_1, V_2 and V_3. Under original schedules, their arrival times at different nodes during their own journey are (T_{1A}, T_{1C}), (T_{2A}, T_{2A}), and (T_{3B}, T_{3C}) correspondingly. It is known that, V_1 conflicts with V_2 at A, thus V_1 needs to be delayed for d_{11}; V_2 conflicts with V_3 at B, thus V_2 needs to be delayed for d_{21}; and V_1 conflicts with V_3 at C, thus V_3 needs to be delayed for d_{31}. The target is to find out a resolution to reschedule the vessels, so that under the renewed schedules, no conflict occurs between each pair of vessels.

Suppose that in the renewed schedules,

d_1 : the total delay of V_1,
d_2 : the total delay of V_2,
d_3 : the total delay of V_3.

To avoid the conflicts between V_1 & V_2, V_2 & V_3 and V_1 & V_3, the final delays for the vessels must satisfy

The delay on V_1 should be bigger than the delay on V_2 : $d_1 - d_2 \geqslant d_{11}$.
The delay on V_2 should be bigger than the delay on V_3 : $d_2 - d_3 \geqslant d_{21}$.
The delay on V_3 should be bigger than the delay on V_1 : $d_3 - d_1 \geqslant d_{31}$.

The above equations give

$$0 \geqslant d_{11} + d_{21} + d_{31}.$$

This leads to

$$0 = d_{21} = d_{31} = d_{11}.$$

It is to say, under existed conditions, no resolution exists to eliminate all the conflicts. Thus, an infinite loop is bound to occur in the node-based preplanning process.

According to the mathematical analysis of the infinite loop, the node-based preplanning is applied only locally and separately at each node. The vessel with later arrival time will be delayed. In other words, the priority of each vessel is associated with its arrival time. A vessel with lower priority is delayed to avoid the potential conflict. As such, for a pair of vessels, one may have higher priority at certain node but lower priority at another node. This is in fact a limitation to eliminate conflicts. If admitting that the vessel ahead in the conflict situation can be delayed, the equations would have solutions. For example, V_1 conflicts with V_3 at C. V_1 can be delayed for d_{13}. Thus, the new equations become

$$d_1 - d_2 \geqslant d_{11},$$
$$d_2 - d_3 \geqslant d_{21},$$
$$d_1 - d_3 \geqslant d_{13},$$

The solution is,

$$d_2 = d_3 + d_{21},$$
$$d_1 = \max(d_3 + d_{13}, d_3 + d_{11} + d_{21}).$$

Following the same way, another solution with minimum delay is,

$$d_3 = 0,$$
$$d_2 = d_{21},$$
$$d_1 = \max(d_{13}, d_{11}, d_{21}).$$

The problem of an infinite loop can thus be resolved. In the real conflict situation, it may not be reasonable to require the vessel ahead to give way to the vessel behind it, due to the principle of FCFS. However, in the preplanning stage, the way to delay the vessel ahead in a conflict situation does not violate the principle of FCFS. For example, a conflict occurs between V_1 and V_2, where V_1 is ahead of V_2. With previous method to eliminate this conflict, delay V_2 and remain V_1 unchanged. Note that, when a vessel is delayed at some time in the preplanning process, its movements during its whole journey will change. Each time a vessel is adjusted, one cannot predict the change of conflict situations in the entire network. Thus, although V_2 is delayed at current moment, it is uncertain that at the end of preplanning, the final delay for V_1 will be less than the final delay for V_2. Likewise, in this example if delaying V_1 and remaining V_2 unchanged, it does not imply that the final delay for V_1 will be larger than that of V_2. Therefore, for a single conflict between two vessels, it is theoretically feasible to adjust any vessel to eliminate the conflict.

5.2.4 Vessel-based rescheduling

A new algorithm will be proposed, which performs the rescheduling process vessel-by-vessel. Vessel-based rescheduling algorithm would overcome the limitations in the node-based algorithm, i.e. avoid the infinite loops and thus enable a feasible solution for conflict-free rescheduling.

5.2.4.1 Algorithm

Figure 5.12 is the flowchart of the vessel-based rescheduling algorithm. The first step is to set up the arrival time table on each node in the traffic network. Different from the node-based rescheduling algorithm, which lists arrival times for all vessels in a single time table, vessel-based rescheduling arranges an arrival time table on each node following the notations in Section 5.2.1.6. The arrival time table (see Table 5.1) contains more details than Table 5.2 used for node-based rescheduling, which is helpful to check both link conflicts and node conflicts.

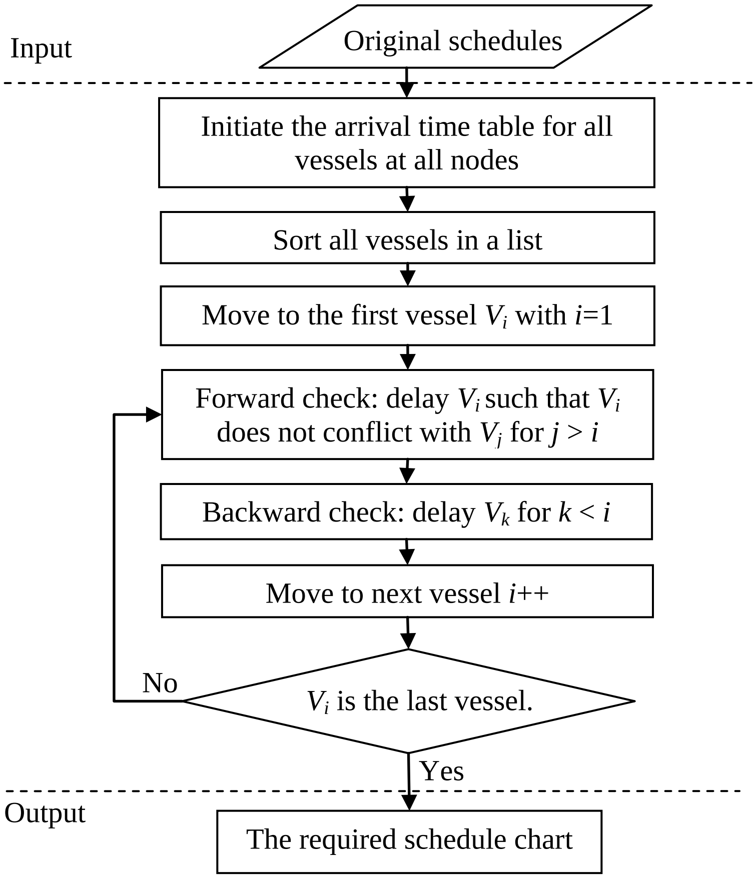


Figure 5.12 The flowchart of vessel-based preplanning.

The second step is to sort all vessels in the priority list in ascent order (see Section 5.2.1.4). During the rescheduling process, a vessel with lowest priority will be delayed firstly. It is assumed that for $i < j$, V_i has lower priority than V_j.

The third step is to check each vessel one-by-one in the priority list. The checking for each vessel contains two parts:

- Forward check: check the time interval between V_i and V_j for $j > i$; delay V_i such that V_i does not conflict with V_j.

- Backward check: check the time interval between V_i and V_k for $k < i$ using the sequence $k = i-1, i-2, \dots, 1$; delay V_k such that V_k does not conflict with V_i.

Forward check

Suppose a vessel is selected for rescheduling, denoted as V_s. During its traveling in the traffic network, V_s may be involved in potential node conflicts and link conflicts with other vessels. When it conflicts with any vessel with higher priority, we will delay the arrival times of V_s so as to avoid the conflict. Suppose

a : the number of node conflicts that the vessel is involved in,
b : the number of link conflicts that the vessel is involved in,
d_{1i} : the required delay for the vessel in the i-th node conflict, and
d_{2j} : the required delay for the vessel in the j-th link conflict.

If the priority of V_s is higher than the other vessel in the conflict, the delay value, d_{1i} or d_{2j}, will be set to be zero. Otherwise, Eq.(5.11) will be adopted to calculate d_{1i} and Eq.(5.13) will be adopted for d_{2j}. The total delay for V_s to avoid all potential conflicts will be

$$d = \max_{\substack{i=1,2,\cdots,a \\ j=1,2,\cdots,b}} \left\{ d_{1i}, d_{2j} \right\}. \tag{5.20}$$

The above delay will be added to original arrival time of V_s, which ensures that all previous $(a+b)$ conflicts are eliminated. However, adjustment of the arrival time of V_s may introduce new conflicts. For example, V_s conflicts with V_2 at node A and V_s conflicts with V_3 at link B. After applying the delay, it is possible that V_s conflicts with V_2 at link B and V_s conflicts with V_3 at node A. Therefore, we need to repeatedly adjust the arrival time of V_s with delays until it is not in conflict with any vessel with higher priority.

Backward check

Forward check is to eliminate conflicts between the selected vessel and the vessels with higher priority. However, because V_s is delayed, it may conflict with the vessels with lower priorities. Then, a backward check needs to be executed for all these vessels V_k ($k < s$). Starting from V_{s-1}, we will perform the forward check again and update its arrival time. The backward check is an iterative process which executes forward check repeatedly from V_{s-1} to V_1.

Suppose the number of vessels in the system is n. After applying the forward check for the i-th vessel, the vessel-based rescheduling can guarantee that

- It will not conflict with the j-th vessel for $j > i$.
- Additional (i-1) forward checks on those vessels with lower priorities than the priority of V_i will be applied during backward check.

Thus for each vessel V_i, the number of forward checks is i. Therefore, in vessel-based rescheduling, the total number of forward checks is

$$\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}. \tag{5.21}$$

Compared with the node-based rescheduling, the vessel-based algorithm has two advantages. Firstly, the rescheduling process can always finish in finite steps (Eq.(5.21)), and thus guarantees a solution. Secondly, the rescheduling process is conducted exactly following the order of vessel priorities. It starts by adjusting schedule of the vessel with lowest priority and finishes by checking the vessel with second highest priority. The schedule of the vessel with highest priority keeps unchanged.

5.2.4.2 Rules in preplanning

Priority FCFS

A vessel before arriving will inform the seaport, for the purpose of ordering a berthing/anchorage area for loading/unloading or ordering services for maintenance, piloting, and refueling. Thus, vessel appearing earlier in schedules, has a higher priority, which is known as the first-come-first-serve (FCFS) rule.

During the preplanning stage, vessel priority is taken into account in conflict resolution. It means that, to resolve a conflict between two successive vessels, necessary delay should be added into the schedule of the vessel whose priority is lower. At present, the system always delays the vessel behind as a default setting. The priority of each vessel can be set in the simulation system. For example, if bid successfully in port, a vessel may be upgraded to higher priority before the preplanning.

Delay tolerance for a vessel in preplanning

In theory, a certain amount of delay in one vessel schedule is enough to eliminate a conflict. However, when the delay is very huge, it is obviously unacceptable and unreasonable in real world. Delay tolerance is the limit on delay for a vessel that is acceptable, which can be set in the simulation system.

If the required delay in a vessel’s schedule is beyond its delay tolerance, the system has to ask for other ways to resolve the conflict. Delay tolerance for a vessel can vary, according to fairway characteristics, traffic regulation, and the urgency of the demand it requires.

Paralleling travel in a link

Even through the time interval between two vessels are inadequate during a link, a conflict will not occur as long as the link width is sufficient. For example, a vessel V_1 follows the other vessel V_2 in a link (Figure 5.13), where $v_1 > v_2$. Previous method for conflict detection is merely based on an assessment of the time interval between the two vessels. If the time interval is inadequate, the system would judge a conflict will occur. Considering the link width, the result may be different. Suppose the width of the link is W, and the width of the vessels’ domain are W_1 and W_2. The two vessels can travel paralleling without a conflict under the condition that $W \geq W_1 + W_2$.

Figure 5.13 Paralleling travel.

Therefore, procedure of conflict prediction can be divided into two steps: to assess time interval, and then check whether vessels can travel paralleling. However, if several vessels may travel paralleling in a link, the situation will be quite complicated.

Suppose three vessels travel along a same link, where $v_1 > v_2 > v_3$. Meanwhile, vessel V_3 comes into the link first, followed by V_2, and V_1 is the last. If link width (W) and the widths of the vessels’ domains (W_1, W_2, W_3) satisfy the following equations,

$$W_1 + W_2 + W_3 > W \geq W_1 + W_2, W_1 + W_3, W_3 + W_2 .$$

It means that, the three vessels are not allowed to travel paralleling together, but any pair of them can travel paralleling. During the preplanning, the assessment of time intervals is always made for two vessels. In this case, the system may possibly judge no conflict will occur in the link, because each pair of the three vessels is allowed to travel paralleling. An algorithm is hereby developed, through an improved assessment of time intervals, to check whether the three vessels can travel paralleling simultaneously. If a conflict occurs, the system will delay V_1, who has lower priority compared to others.

Firstly, one can derive the following time intervals:

$[A_{12}, B_{12}]$, when V_1 travels paralleling with V_2,

$[A_{13}, B_{13}]$, when V_1 travels paralleling with V_3,

$[A_{23}, B_{23}]$, when V_2 travels paralleling with V_3,

where

A_{ij}: the beginning time when V_i travels paralleling with V_j,

B_{ij}: the ending time when V_i travels paralleling with V_j.

These intervals are denoted as paralleling intervals. If V_1, V_2 and V_3 can travel paralleling together, a paralleling interval $[A_{123}, B_{123}]$ must exist,

$$[A_{123}, B_{123}] = [A_{12}, B_{12}] \cap [A_{13}, B_{13}] \cap [A_{23}, B_{23}].$$

As long as $[A_{123}, B_{123}] \neq \varnothing$, it indicates that the three vessels cannot travel paralleling together. Therefore, in order to judge whether *n* vessels ($V_1, V_2, ..., V_n$) can travel paralleling, there are paralleling intervals $[A_{ij}, B_{ij}]$ for all pairs of vessels v_i and v_j. These vessels cannot travel paralleling together, if

$$\bigcap_{i,j} [A_{ij}, B_{ij}] = \varnothing .$$

Otherwise, the vessel with lowest priority will be delayed. Suppose two vessels travel along a link from node *A* to node *B*, whose length is *M*. Let

V_1 arrives at *A* at t_1, and arrives at *B* at $t_1^{'}$,

V_2 arrives at *A* at t_2, and arrives at *B* at $t_2^{'}$,

$t_1 < t_2$, i.e. V_1 reaches *A* before V_2.

It is known that, V_2 will overtake V_1, if and only if $t_2^{'} > t_1^{'}$ i.e. V_2 reaches *B* before V_1. Thus

$$(t_2 - t_1 + r) \cdot v_1 - S_{12} = r \cdot v_2,$$
$$(t_2 - t_1 + s) \cdot v_1 + S_{12} = s \cdot v_2, \ S_{12} = \frac{L_1 + L_2}{2},$$

which leads to

$$r = \frac{(t_2 - t_1)\cdot v_1 - S_{12}}{v_2 - v_1},$$

$$s = \frac{(t_2 - t_1)\cdot v_1 + S_{12}}{v_2 - v_1},$$

where

v_i : the speed of *V_1*,

S_{ij} : the minimum distance separation between *V_i* and *V_j*,

L_i : the length of the domain of *V_i*,

r : the time v_2 catch v_1 after v_2 pass *A*,

s : the time v_2 overtakes v_1 after v_2 pass *A*.

Thus, the paralleling interval between V_1 and V_2 is

$$[A_{12}, B_{12}] = [t_2 + r, t_2 + s].$$

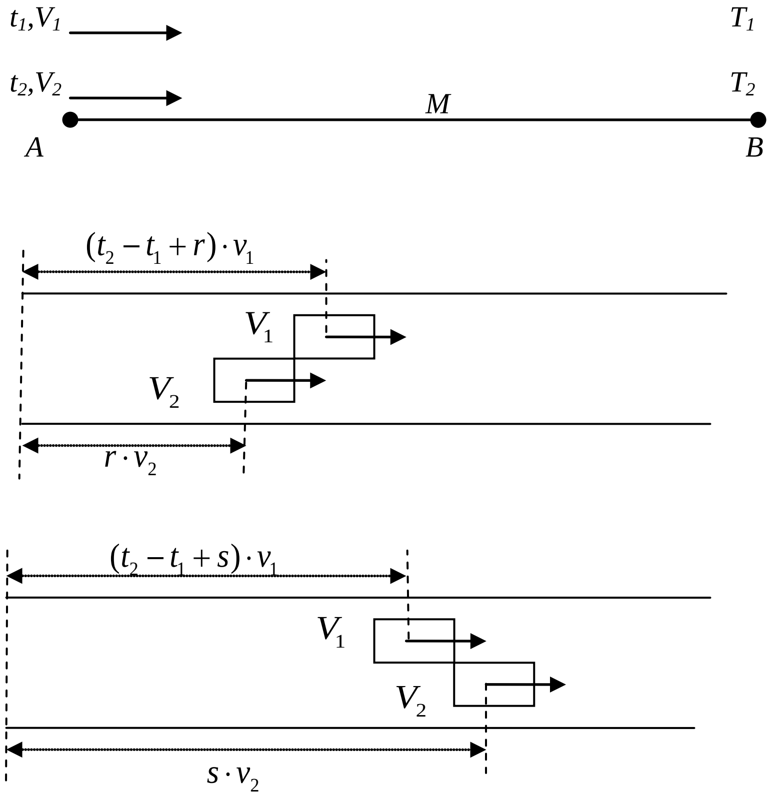


Figure 5.14 Calculation for paralleling interval.

5.2.4.3 An alternative algorithm

Figure 5.15 presents the flowchart of an alternative algorithm for Figure 5.12. The alternative algorithm has advantages as follows,

- It performs the forward check for (n-1) times without backward check. It is faster than the original algorithm in Figure 5.12.
- It produces a better result than the original algorithm in Figure 5.12 in the sense of minimizing the total delay for all vessels.

A simple example can be presented to illustrate the performance of delay minimization. Suppose three vessels in the order of priorities as V_1, V_2, and V_3. V_1 and V_2 conflict, and while V_2 and V_3 conflict. The result from the original algorithm generates a delay d_1 for V_1 and a delay d_2 for V_2. However, use of the new algorithm may provide only a delay d_3 for V_2. This means that all conflicts are removed through delaying one vessel V_2 and it is possible that $d_3 < d_1 + d_2$. However, one drawback of this algorithm is that it does not follow the order of vessel priorities. In the preplanning stage, both algorithms are provided for users in the simulation system.

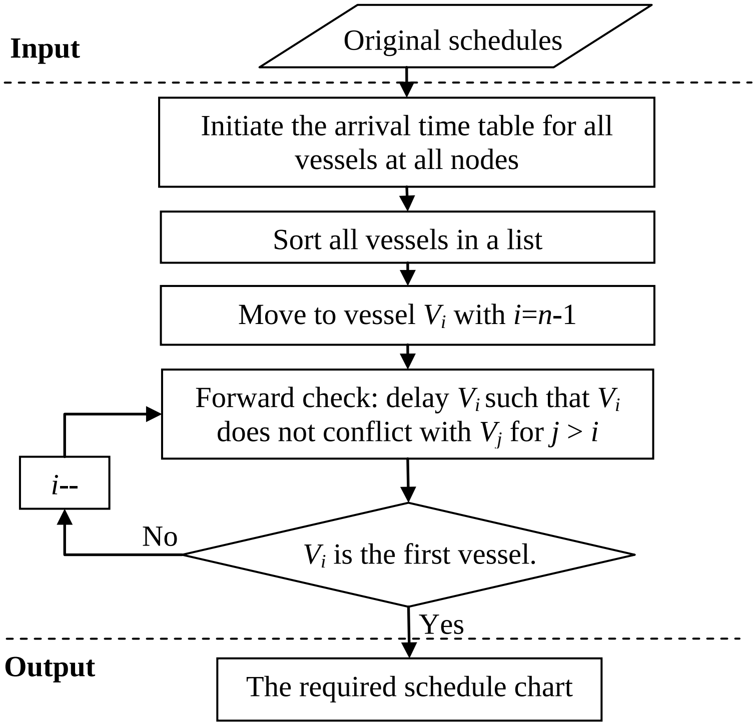


Figure 5.15 The flowchart.

5.2.4.4 Delay minimization

Given the initial schedule chart S^0 in Eq.(5.1), the solution of finding a new schedule chart satisfying the minimum time separation is not unique. As presented in the previous section, given a priority list

Ψ in Eq.(5.8), the vessel-based rescheduling can provide a result guaranteeing the minimum time separation. However, as discussed in Section 5.2.1.4, for *n* vessels, the number of different priority lists can be up to $n!$. Thus, given a schedule chart, there is up to $n!$ number of result schedule charts. This leads to an optimization problem: how to find the schedule chart $\overline{S}$ that minimizes the total delay for all vessels, i.e.

$$\begin{aligned} &\min : \delta = \operatorname{sum}\left(\Delta\right) = \sum_{i=1}^{n} \delta_i, \\ &\text{with} : \Delta = \overline{S} - S^0, \end{aligned} \tag{5.22}$$

where Δ includes all the delays for all vessels as defined in Eq.(5.3).

One solution to find the optimized schedule chart (Eq.(5.22)) is to find out all feasible schedule charts and acquire the one with the minimum delays. However, as the number of vessels (*n*) increases, the total number of priority list ($n!$) will dramatically increases. Finding the solution by using all the priority list cannot work since it is

- impossible: normally the number of vessels is huge,
- unnecessary: different priority lists may lead to a same schedule chart.

Take $n = 4$ as an example. Table 5.13 lists the priority lists where vessel 4 has the highest priority. The right column lists the vessel sequence to apply forward check. Actually, different vessel sequences will lead to different solutions. The *n*! priority lists do not cover all the vessel sequences. For example, for vessel 4 has the highest priority, the vessel sequence [1, 2, 3, 1] is not covered. It is possible that the optimized solution lies within the uncovered vessel sequences.

Table 5.13 Some priority lists for *n* = 4.

Priority list	Vessel sequence for forward check
[1,2,3,4]	[1,2,1,3,2,1]
[1,3,2,4]	[1,3,1,2,3,1]
[2,1,3,4]	[2,1,2,3,1,2]
[3,1,2,4]	[3,1,3,2,1,3]
[2,3,1,4]	[2,3,2,1,3,2]
[3,2,1,4]	[3,2,3,1,2,3]

The global optimized solution (Eq.(5.22)) can be obtained by the means of schedule tree (definition see Section 5.2.1.5), which include all possible vessel sequences. And a local minimization can be achieved by controlling the maximum level of the tree structure.

The schedule tree for vessel-based rescheduling can be set up as follow

- Root node: the tree root is the initial schedule chart S^0 (Figure 5.16).
- Children node: in the schedule chart N_i at tree level l, with other vessels' schedules unchanged, if the *j*-th vessel V_j needs a delay to guarantee the minimum time separation with other vessels, a tree node $N_{i,j}$ is created at tree level *l* + 1 as a child of N_i.
- Repeated node removal: after creating a tree node $N_{i,j}$, the algorithm will check whether it is equivalent with any existing tree nodes. If yes, this tree node $N_{i,j}$ is removed.
- Maximum tree level: the tree is built up to the maximum tree level, which is an input from the user.

For example, in Figure 5.16, from the root S^0,

Delaying only V_1 while keeping others unchanged lead to the child node N_1,

Delaying only V_2 while keeping others unchanged lead to the child node N_2, and

Delaying only V_3 while keeping others unchanged lead to the child node N_3,

Similarly, from the tree node N_1,

Delaying only V_2 while keeping others unchanged lead to the child node $N_{1,2}$, and

Delaying only V_3 while keeping others unchanged lead to the child node $N_{1,3}$,

The tree can be built up to a prescribed tree level. In our system, by default, the tree level is taken as the number of vessels. As a result, the tree leaf node can be of any of the following

- A feasible schedule chart guaranteeing all minimum time separations,
- A repeated tree node that is not a feasible schedule chart, and
- A tree node at the maximum tree level that is not a feasible schedule.

To save memory for computing, the tree structure in Figure 5.6 can be adopted by removing the intermediary tree nodes.

Upon the schedule tree, it is feasible to traverse all the feasible schedule charts and find out a schedule chart with minimal total delay. This produces a local minimal to the optimization problem. Moreover, solving the global optimized solution (Eq.(5.22)) may not be enough. In a real-life seaport, more other factors need to be taken into account. If all the delay accurse in a small local region, it is likely to introduce traffic congestion (See Section 5.3.5 for solutions).

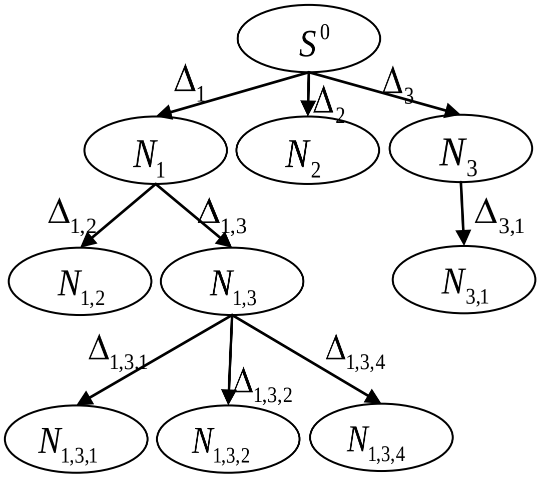


Figure 5.16 A schedule tree for vessel-based preplanning.

5.2.5 Examples and discussions

Preplanning at a node

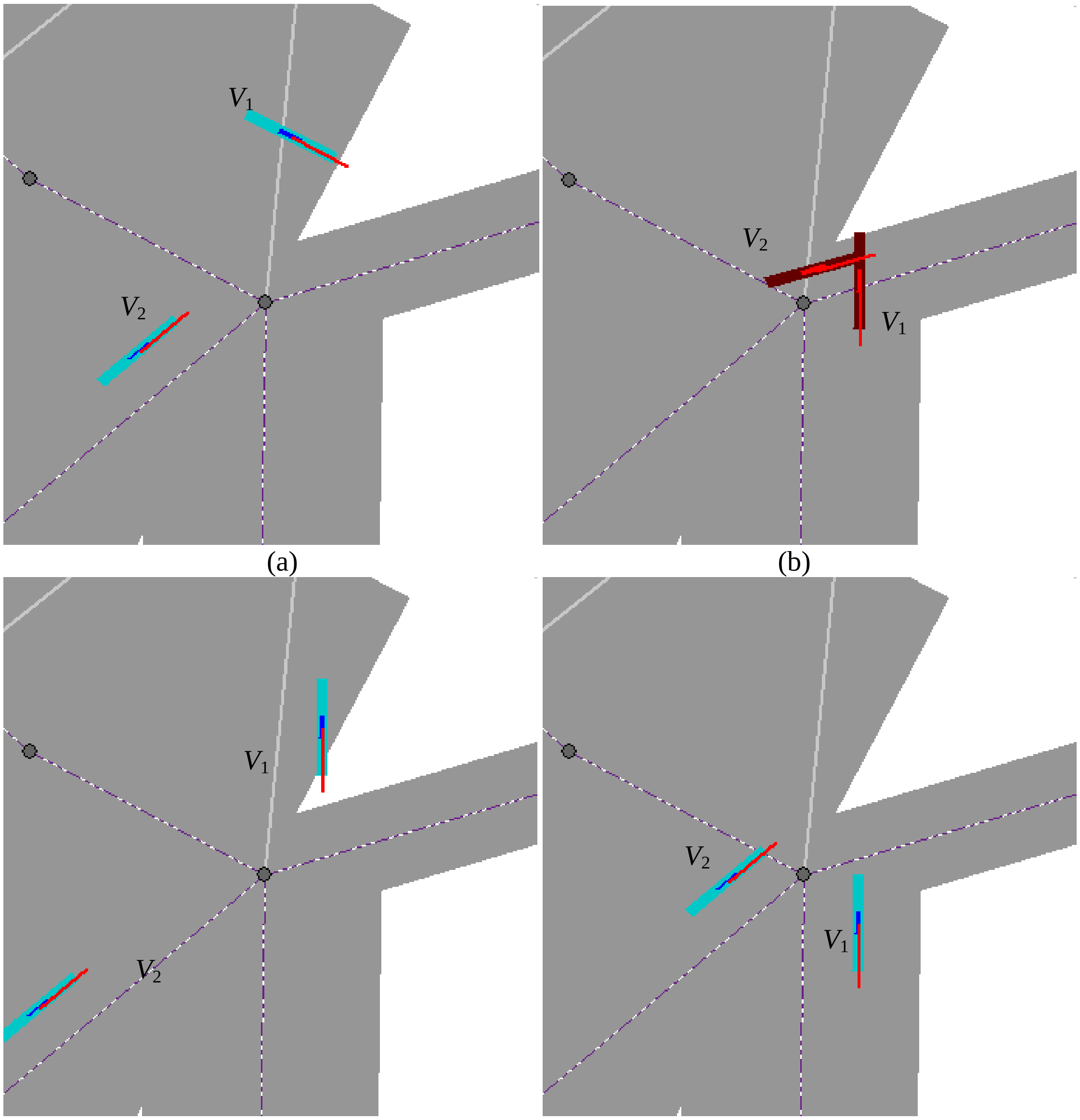
The two vessels are crossing with each other at a node. Details of the two vessels are listed in Table 5.14. By default, V_1 has higher priority, thus, V_2 has to be delayed in the preplanning if the minimum time separation is not satisfied between them.

Table 5.14 Details for two vessels at a cross situation.

Vessels	V_1,V_2
Speed	12 knots
LOA	90m
Beam	18m

Figure 5.17 shows how a conflict is resolved between the two vessels around the node. The two vessels in Figure 5.17(a, c) follow the original schedules and Figure 5.17(b, d) show the result after preplanning. In Figure 5.17(a), V_1 and V_2 are approaching to the same node, and a conflict occurs between them in Figure 5.17(c). During the preplanning process, V_2 is delayed for 139 seconds in total. After preplanning, V_2 is pushed back for V_1 can pass through safely, as shown in Figure 5.17(b, d).

In algorithm design, a node in the network is treated as a point to estimate the arrival time of a vessel. However, a node displayed in screen is an area connecting links. Vessels in the simulation system travel along the medial line of fairways. Thus, a vessel turning at a corner may not exactly reach the point. In the Figure 5.17(c), both vessels do not pass through the center of the node. A small tolerance is hereby resulted in the estimated arrival time; likewise, delay derived from preplanning algorithm also has a small tolerance.



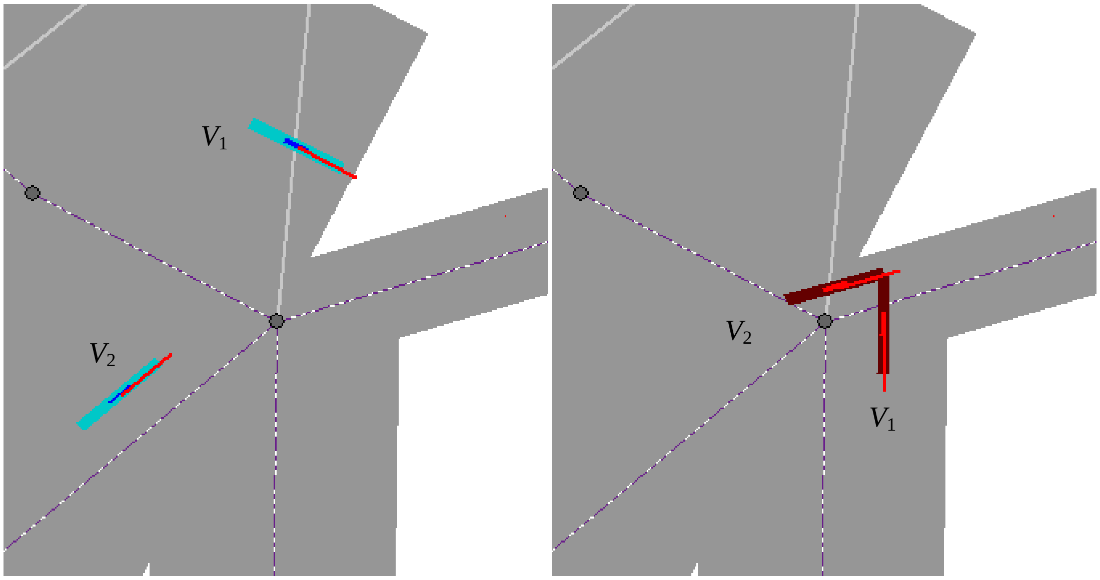
(a) (b)

(c) (d)

Figure 5.17 Preplanning result around a node.
(a,b) two vessels with original schedules, and (c,d) the result after preplanning.

<u>Preplanning with delay tolerance</u>

In the previous example, V_2 is delayed for 139 seconds. Delay tolerances can be applied to vessels. Suppose the delay tolerance of V_2 is 120 seconds. Then, there is an irresolvable conflict (Figure 5.18). Otherwise, if the delay tolerance of V_2 is bigger than 139 seconds, this conflict can be resolved. In Figure 5.18(b), after preplanning, due to the delay tolerance, the minimum time separation is not guaranteed and there is a conflict between the two vessels. However, as implemented in Chapter 4, in the simulation stage, this conflict will be predicted far away before these two vessels reach this node. Thus, there will be enough time for V_2 to take actions (make turns or decelerate) to avoid this conflict.



(a) (b)

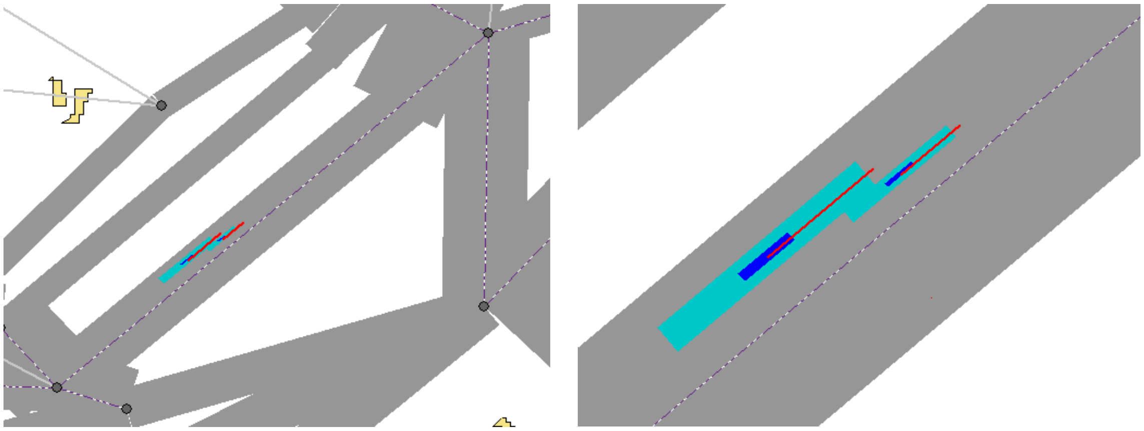
Figure 5.18 The preplanning result with delay tolerance.

(a) two vessels approaching a same node, and (b) two vessels conflict.

As discussed previously, preplanning aims to provide a reasonable schedule chart. Removing all conflicts to guarantee safety navigation is the task for the next stage: simulation. Without preplanning, the behaviors for different vessels are unpredictable. There may be hundreds of conflicts in the initial schedule chart. Leaving all conflicts for the simulation stage is unreasonable and unacceptable. This may cause serious troubles. The preplanning can help to reduce the number of conflicts. Each conflict will also only like the case in Figure 5.18(b), where V_2 only needs to slightly decelerate or make a small turn to avoid this conflict.

<u>Preplanning in a link</u>

The preplanning also takes into account the width of a link. If link width is sufficient, vessels can travel paralleling without conflict. In this example, two vessels travel paralleling on a same link (Figure 5.19). However, two vessels will fail to travel in parallel on a same link if the link width is not sufficient. In such case, preplanning will delay the vessel with lower priority.



(a) (b)

Figure 5.19 Two vessels paralleling on a link.
(a) a vessel is over taking the other vessel, and (b) the enlarged image.

<u>Preplanning for a group of vessels</u>

In this example, the simulation system makes the preplanning for a traffic system of 30 vessels. All the vessels are selected randomly by the system of different vessel types. Moreover, the system automatically specifies a route in the network for each vessel, consisting of nodes and links connected with each other. The priority of FCFS is applied in conflict resolution. All vessels have the same delay tolerance 180 seconds.

Figure 5.20 shows two screenshots during the simulation of the original schedule chart. The total number of conflicts under the original schedules is 86. After the preplanning, the total number of conflicts is reduced to 9. The preplanning cannot eliminate all the conflicts due to:

- The arrival time of a vessel at a node is an estimated value, because the preplanning treats each node as a point and calculates the arrival time as the time it reaches the point. However, during simulation, the vessel may not pass the point. Thus, there is a tolerance in arrival time.
- Each vessel has its own delay tolerance. If the delay required is beyond the delay tolerance, the vessel will be delayed for the maximum acceptable delay (i.e. delay tolerance). In these cases, a conflict remains irresolvable.

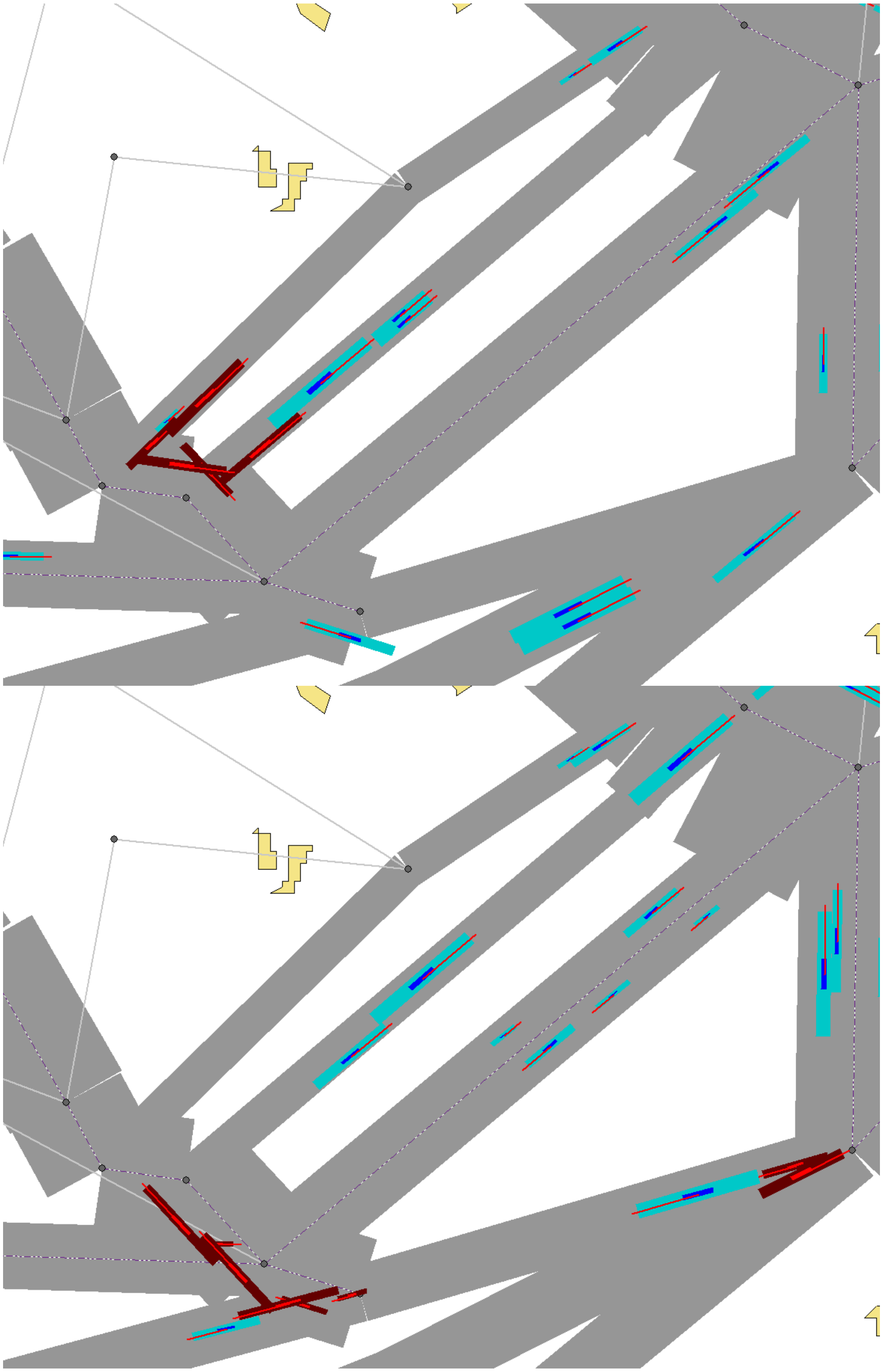


Figure 5.20 Two screen snap shots.

Preplanning times

The simulation system is developed using Visual C++. The above example shown in Figure 5.20 is tested on different PCs to check the running times in the preplanning stage. On each PC, 30 trials are tested. Each trial performs preplanning on the initial schedule chart. Of course, all trials will produce a same preplanning result. For the running time, different trials only have at most 1 millisecond difference.

The running for the two PCs is listed in Table 5.15. According to data collected in the table, the time-driven vessel-based preplaning is very fast. Acutally, after testing on different schedule charts on above two PCs, one minute is enough for preplanning hundreds of vessels.

Compared with the time-driven vessel-based preplaning, the event-driven vessel-based preplanning is time consuming. Time-driven vessel-based preplaning does not require the running of the simulation. It only requires the calculation of arrival times of each vessels at all nodes/links. On the other hand, the event-driven vessel-based preplanning consists of two steps:

Step 1: Run one round of simulation to detect all conflicts. If no conflict is detect, finish. Otherwise go to Step 2.

Step 2: Select one vessel. Eleminate all conflicts that this vessel involves. Derive a new schedule chart. Continue with Step 1.

As such, since the example shown in Figure 5.20 contains 30 vessels and 86 conflicts. Resolving all conflicts requires 29 rounds of simulation and it takes about 30 minutes to finish the event-driven vessel-based preplanning, which costs much time than the time-driven vessel-based preplanning.

Table 5.15 Running time of preplanning.

PC configuration	Total milliseconds
Operation system: Windows XP CPU: Intel(R) Core(TM) 2 Duo CPU T8100@2.10GHz RAM: 1.96G	109
Operation system: Windows XP CPU: Intel(R) Core(TM) 2 Duo CPU E8400@3.00GHz RAM: 3.25G	47

5.3 The simulation stage

The preplanning stage can resolve potential conflicts under original vessel schedules. However, there will be fluctuations in the schedule for each vessel. For example, the actual arrival time of each vessel may be different from the pre-assigned schedule, and the actual time for each vessel to leave the seaport may also vary due to the uncertainty in the loading/unloading working time. Conflicts resulting from these random fluctuations will be considered in the simulation stage.

For an individual vessel, the most direct way to resolve a conflict is to take evasive maneuvers. However, few port authorities actually control vessel maneuvers once the vessel's routing has been approved. It is hardly possible, in a real encounter, to consider so many variables with expert knowledge to come up with an optimal solution for evasive maneuvers. In the sense of traffic control, a necessary measure is to avoid imminent conflict situations which are urgent events or emergencies. A multi-link, look ahead conflict prediction will be executed a few links in advance, thus allowing sufficient time for navigators to take evasive actions. The applicable corrective measures are identified by a decision-making mechanism. These measures conclude general rules or guidance for navigator to choose proper evasive maneuvers, but how the actual maneuvers are to be conducted is not of significance in this regard.

In this section, the simulation stage is implemented

- To check whether the preplanning result is acceptable,
- To predict conflict risks due to random vessel movements, and
- To provide possible corrective measures to resolve such conflicts.

5.3.1 Preliminaries

During the simulation stage, multi-link look-ahead conflict prediction is executed to predict a conflict risk between any two vessels before they actually encounter. The number of links required with look-ahead can be set up in the simulation system. Theoretically, there is sufficient time for vessels to take corrective actions to avoid the conflict.

Corrective actions proposed here include course alteration and deceleration. In the simulation system, each vessel has its own parameters to take these actions, including

- Max turn angle α, which is the maximum angle that the vessel can make turns. Usually, this parameter is pre-decided after it was build. Operators can reduce the value for this parameter, then, under same conditions, the vessel will need to take turning action earlier.
- Decelerated rate r, which decides the minimal speed in the decelerating procedures. If the speed before deceleration is v, then, during the deceleration, the minimal speed should not smaller than vr. Normally, $r \in (0,1)$. Similar with maximum turn angle, if operators reduce the value for this parameter, under same conditions, the vessel will need to take decelerate

earlier in order to avoid a same conflict. Usually, during a deceleration, vessel is prevented from stopping, i.e. the minimal speed reach 0, or $r = 0$.

To control the turning and decelerating abilities for each vessel, the following parameters are introduced in our simulation system:

- Multi-link-ahead parameter m, which is introduced in Section 4.2.2.4 indicating the number of links ahead current vessel position to perform the conflict prediction. Obviously, bigger value gives operators more time to take actions.
- Expected turn distance l, which indicates the minimal distance to the meeting point where the vessel needs to initialize the turning. In real life situation, the closer the vessel to the meeting point, the bigger turning angle it requires to avoid the conflict. Thus this value should not be too small. On the other hand, it is unacceptable for a vessel to make turns too far away from the meeting point. Thus, a proper turn distance l is important for each vessel. In our system, by default, the l value for each vessel V_i is taken as

  $$l_i = \eta L_i$$

  where L_i is the length of the domain for V_i, and η is a global parameter applied for all vessel. The default value for η is 2. Operator can assign the value for each separate vessel.

5.3.2 Decision-making for corrective actions

Figure 5.21 is the flowchart of decision-making for corrective measure for conflict resolution. After a conflict is predicted, the first step is to check the type of the conflict, i.e. to check whether it is a node conflict or a link conflict. Decision making of a proper action is based on a rule that: if a conflict is predicted in a link, the overtaking action is taken into account once it is allowable; otherwise, a vessel that is indentified as give-way vessel will decelerate so as to avoid conflicting with the other vessel. In real life, the role of give-way vessel is dependent on encounter situation and specialized traffic rules. In this research, a vessel with lower priority is treated as the give-way vessel. Vessel priority is an input parameter controlled by the simulation system.

For conflict resolution on a node, it is inefficient for vessels to decelerate because of more delays will result in. Thus, course alteration is taken the first priority. However, the course alteration of a vessel will affect other vessels and may result in risk of conflicts. A turning trajectory is first needed to be derived. The system then predicts whether the turning vessel will conflict with other vessels during the turning process. If a potential conflict is predicted, this vessel is not allowed to change course but to decelerate.

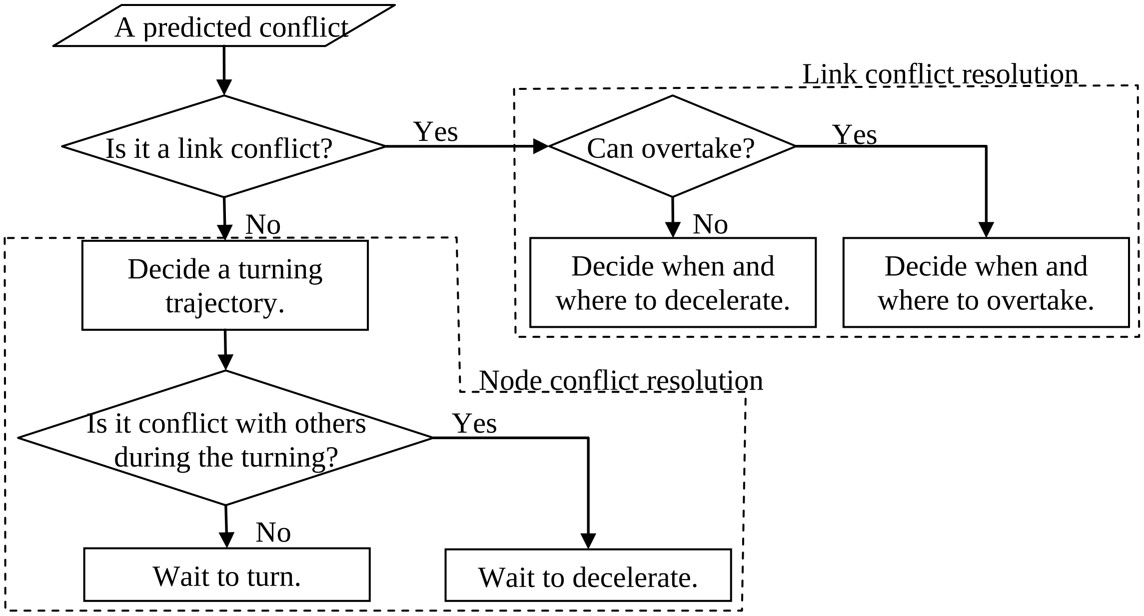


Figure 5.21 The flowchart of conflict resolution to avoid a predicted conflict.

It is noticed that the turning trajectory for conflict avoidance is not unique. Considerations of different factors will lead to different results. In the following section, a simple way to calculate a feasible turning trajectory is presented. Similarly, once a vessel decides to decelerate, it also need to decide when and where to decelerate. Such resolution is also not unique. In the simulation system, some simple rules have been defined for vessels to decelerate. For a link conflict, a vessel can decelerate so as to keep the minimum distance separation with the vessel ahead. Its speed will be recovered after it leaves the link. For a node conflict, deceleration of a vessel must ensure that another crossing vessel can pass the node safely. The decelerated vessel will recover speed only after it passes the node.

The following two sections will address the following problems. Some simple cases are considered to solve the problems.

- To resolve a node conflict, how to obtain a turning trajectory for the vessel? If the vessel cannot take the turning trajectory, when and where the vessel should decelerate?
- To resolve a link conflict, how to check whether the vessel can overtake the vessel ahead or not? If not, when and where it should decelerate?

5.3.3 Node conflict resolution

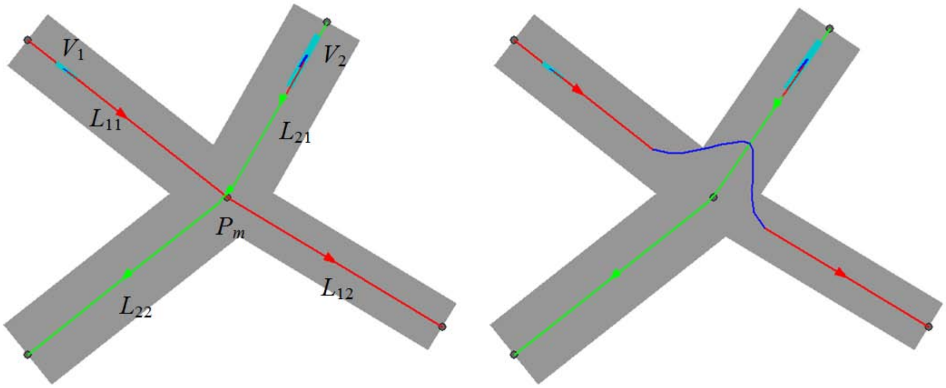
Suppose the two vessels V_1 and V_2 in Figure 5.22(a) are traveling toward a same node, and a conflict is predicted near the node. Table 5.16 collects the navigation information for both vessels. Suppose V_2 reaches the meeting point P_m before V_1. Thus, if V_1 conflicts with V_2, V_1 needs to make a turn or decelerate (Figure 5.22 (b)). V_1 will make a turn according to the parameters α and l_1. Otherwise, it will decelerate using the parameter r_1.

When a conflict is predicted, the flowchart in Figure 5.23 is used to decide whether V_1 should make a turn or decelerate, where (referring to Figure 5.24)

- P_1 and P_2 : turning points, which means that V_1 will change the path at P_1 and return to its original path at P_2,
- θ : the minimal turning angle if the vessel begin turning at point P_1, which should guarantee that there is not conflict after V_1 makes turns,
- P_3 : the reference point which is also the meeting point after V_1 makes the turn.

There are various ways to obtain P_1, P_2, P_3 and θ. Then, a smooth turning trajectory can be defined accordingly. The blue curve in Figure 5.22 (b) shows the turning trajectory for V_1.

The vessel V_1 is not allowed to conflict with other vessels when it makes the turn to avoid the conflict with V_2. Thus, after obtaining the turning course for V_1, the simulation system needs to check whether V_1 conflicts with any other vessel when it makes the turn. This can be easy achieved by apply the conflict prediction. If a conflict with any other vessel is predicted, V_1 will decelerate rather than make a turn.



(a) (b)

Figure 5.22 Two vessels meet at a cross.
(a) a situation for two vessels conflict, and (b) V_1 needs to make a turn to avoid the conflict.

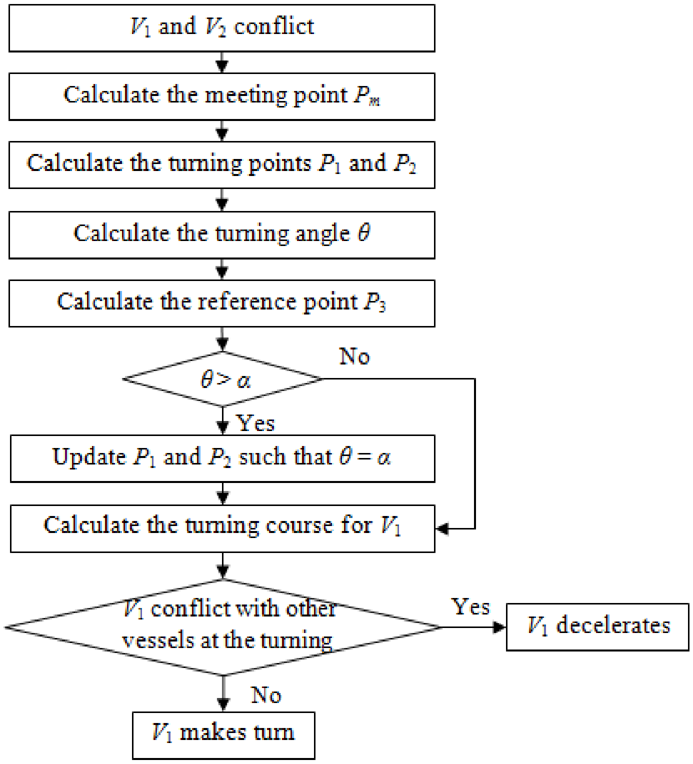
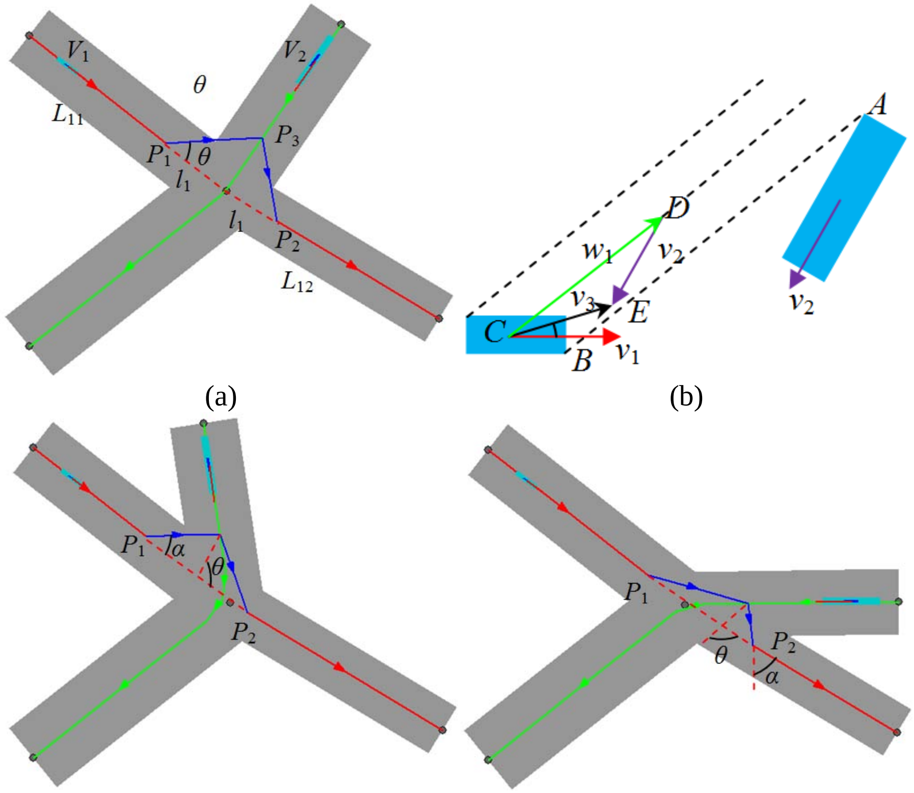


Figure 5.23 The folowchart for V_1 to make a turn or to decelerate for a node conflict.



(a) (b)

(c) (d)

Figure 5.24 Calculate the turning points.

(a) valid turning points, (b) calculate the turning angle θ, (c) update the invalid turning points P_1, and (d) update the invalid turning points P_2.

Table 5.16 Two vessels at a cross situation.

Vessel	V_1	V_2
Max turn angle	α	β
Expected turn distance	l_1	l_2
Current link	L_{11}	L_{21}
Next link	L_{12}	L_{22}
Current Speed	v_1	v_2
Decelerate rate	r_1	r_2
Max acceleration	a_1	a_2
Domain size	$L_1 \times W_1$	$L_2 \times W_2$

5.3.3.1 V_1 makes a turn

For V_1 to make a turn, it needs to obtain the turning course, which can be calculated using the turning points and the turning angle.

Initialize the turning point P_1 and P_2

Initially, the turning points P_1 is on the link L_{11} and P_2 is on the link L_{12} such that

$$|P_1 - P_m| = |P_2 - P_m| = l_1.$$

The vessel will make a turn at point P_1 and return to the original course at point P_2. At point P_1, the vessel makes turn at the angle θ.

Calculate the turning angle θ

Suppose the vessel's speed after turning is v_3. Note that two corners, *A* on V_2 and *B* on V_1 in Figure 3(b), define a safe relative movement for V_1. Thus, the relative movement w_1=v_3-v_2 should parallel to *AB*. Then, the minimal turning angle can be determinate using

$$c = \pi - d,$$
$$b = \arcsin \frac{|v_2| \sin c}{|v_3|},$$
$$\theta = a - b,$$

where

d: the angle between w_1 and v_2,

b: the angle between w_1 and v_3,

a: the angle between w_1 and v_1.

Calculate the reference point P_3

With the turning angle obtain, the vessel V_1 can safely makes turns. It will reach a point P_3 on the link L_{21} (Figure 5.24(a)), which will be the new meeting point between the two vessels after vessel V_1 changes its path.

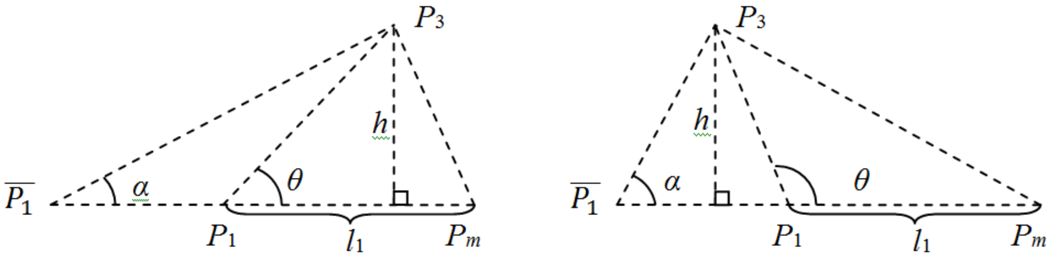
Update P_1 and P_2

If the turning angle excess the maximum value $\theta > \alpha$, the vessel V_1 cannot make the turn (Figure 5.24(c)). Update P_1 to $\overline{P_1}$ such that the new turning angle $\angle P_3\overline{P_1}P_m = \alpha$. Referring to Figure 5.25, the point $\overline{P_1}$ can be calculated by

$$\overline{P_1} = P_m + \frac{P_mP_1}{\|P_mP_1\|}(l_1 + h \cdot \cot\alpha - \|P_mP_1\| \cdot \cos\theta), \tag{5.23}$$

where h is the distance from P_3 to P_1P_m, with $h = \|P_3P_1\| \cdot \sin\theta$.

Eq.(5.23) covers two cases: $\theta \le \pi / 2$ as shown in Figure 5.25(a) and $\theta > \pi / 2$ as shown in Figure 5.25(b). Similarly, if the turning angle on the link L_{12} is too big, the position of P_2 can be updated similarly (Figure 5.24(d)).



(a) (b)

Figure 5.25 Update the turning point P_1.
(a) $\theta \le \pi/2$, and (b) $\theta > \pi/2$.

Calculate turning course

After updating the turning points, the vessel V_1 will makes turn at the point P_1 on the link L_{11}, reach the new meeting point P_3 on the link L_{21}, and finally turn back at the point P_2 on the link L_{12} (Figure 5.24(a)). The path $P_1P_3P_2$ defines a turn for V_1. A smoothing path crossing the three points can be obtained (Figure 5.22(b)).

5.3.3.2 V_1 decelerates

There is an intersect area, Ω, between the motions of the two vessel's domains. In Figure 5.26, the motions of V_1 and V_2 are the red area and the green area, respectively. The intersection area Ω is shown in black. Before decelerate, suppose

t_i : the time when V_i reaches the meeting point P_m ,

$\tilde{t}_i$: the time when V_i enters the intersection area Ω, and

$\bar{t}_i$: the time when V_i leaves the intersection area Ω.

Then,

$$\left.\begin{aligned} \tilde{t}_i &= t_i - \frac{d_i}{v_i}, \\ \bar{t}_i &= t_i + \frac{d_i}{v_i}, \end{aligned}\right. \quad i = 1,2,$$

where $d_1 = \frac{L_1}{2} + \frac{W_2}{2}, d_2 = \frac{L_2}{2} + \frac{W_1}{2}$. Obviously,

$$\bar{t}_1 > t_1 > t_2 > \tilde{t}_2.$$

To avoid the conflict, V_1 needs to decelerate such that V_1 enter the intersect area only when V_2 leave the intersect area. Suppose

P_d: the point where V_1 decelerates,

t_d: the time when V_1 decelerates,

D: the distance from P_d to the meeting point P_m.

v_3: the speed of V_1 after deceleration. v_3 satisfies $v_3 \geq r_1 v_1$, where r_1 is the decelerate parameter.

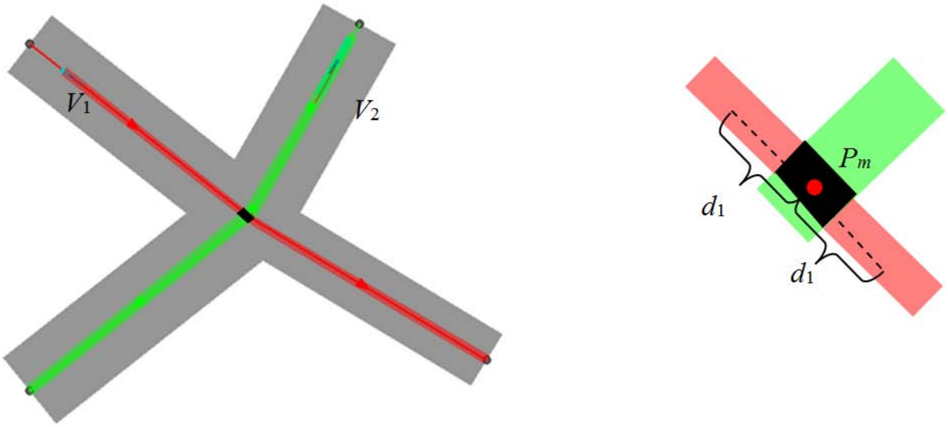


Figure 5.26 The intersection of the motions of the two vessel's domain.

The deceleration of V_1 falls into two categories:

Case 1: If r_1 is small enough, V_1 can keep decelerating until it reaches Ω with $v_3 > r_1 v_1$.

Case 2: Otherwise, V_1 need first decelerates to $v_3 = r_1 v_1$, then travels with v_3 until it reaches Ω.

This means, if V_1 allows reducing a large amount of speed, it can begin decelerating within a short distance to the intersect area. Otherwise, if V_1 only can reduce a small amount of speed, it needs to decelerate far before reach the intersect area.

Theory 5.1: V_1 can decelerate according to Case 1, if and only if

$$1-\sqrt{\frac{2a_1(\bar{t}_2-\tilde{t}_1)}{v_1}} \geq r_1.$$

Otherwise, V_1 will decelerate according to Case 2.

In Case 1, V_1 will decelerate at the time

$$t_d=\bar{t}_2-\sqrt{\frac{2v_1(\bar{t}_2-\tilde{t}_1)}{a_1}}.$$

In Case 2, V_1 will decelerate at the time

$$t_d=\frac{\tilde{t}_1-r_1\bar{t}_2}{1-r_1}-\frac{(1-r_1)v_1}{2a_1}.$$

Proof: For Case 1, the Table 5.17 lists the traveling information for V_1.

Table 5.17 Navigation information for deceleration of Case 1.

	Before deceleration	After deceleration
Speed	v_1	v_3
Time to enter Ω	$\tilde{t}_1$	$\bar{t}_2$
Traveling time from P_d to Ω	D/ v_1	$(v_1- v_3)/ a_1$
Arrival time at P_d	$\tilde{t}_1$ - D/ v_1	$\bar{t}_2$ - $(v_1- v_3)/ a_1$

The arrival time at P_d keeps unchanged, i.e.

$$\tilde{t}_1-\frac{D}{v_1}=\bar{t}_2-\frac{v_1-v_3}{a_1}.$$

Coupling with

$$D=\frac{v_1^2-v_3^2}{2a_1},$$

It gives

$$\tilde{t}_1-\frac{v_1^2-v_3^2}{2a_1v_1}=\bar{t}_2-\frac{v_1-v_3}{a_1}.$$

The above equation can be reformed into

$$v_3^2-2v_1v_3+2a_1v_1\left(\tilde{t}_1-\bar{t}_2+\frac{v_1}{2a_1}\right)=0.$$

Thus,

$$v_3 = v_1 - \sqrt{2a_1 v_1(\bar{t}_2 - \tilde{t}_1)}.$$

The vessel V_1 starts deceleration at the time

$$t_d = \bar{t}_2 - \frac{v_1 - v_3}{a_1} = \bar{t}_2 - \sqrt{\frac{2v_1(\bar{t}_2 - \tilde{t}_1)}{a_1}}.$$

Additionally, since $v_3 \geq r_1 v_1$,

$$\frac{v_3}{v_1} = 1 - \sqrt{\frac{2a_1(\bar{t}_2 - \tilde{t}_1)}{v_1}} \geq r_1.$$

Otherwise, adopt Case 2. For Case 2, Table 5.18 lists the traveling information for V_1.

Table 5.18 Navigation information for deceleration of Case 2.

	Before deceleration	After deceleration
Speed	v_1	$v_3 = r_1 v_1$
Time to enter Ω	$\tilde{t}_1$	$\bar{t}_2$
Traveling time from P_d to Ω	$\tilde{t}_1 - t_d$	$\bar{t}_2 - t_d$
Arrival time at P_d	t_d	t_d

Before deceleration,

$$D = v_1(\tilde{t}_1 - t_d).$$

After deceleration,

$$D = \frac{v_1^2 - v_3^2}{2a_1} + v_3\left(\bar{t}_2 - t_d - \frac{v_1 - v_3}{a_1}\right).$$

Thus

$$v_1(\tilde{t}_1 - t_d) = \frac{v_1^2 - v_3^2}{2a_1} + v_3\left(\bar{t}_2 - t_d - \frac{v_1 - v_3}{a_1}\right).$$

The above equation can be formulated into

$$t_d = \frac{v_1\tilde{t}_1 - v_3\bar{t}_2}{v_1 - v_3} - \frac{v_1 - v_3}{2a_1} = \frac{\tilde{t}_1 - r_1\bar{t}_2}{1 - r_1} - \frac{(1 - r_1)v_1}{2a_1}.$$

5.3.4 Link conflict resolution

The vessel V_1 follows V_2 on a same link (Figure 5.27), and the traveling information is collected in Table 5.19. They may conflict if $v_1 > v_2$. Once a conflict is predicted, V_1 should take actions to avoid the conflict. There are two possible actions: V_1 overtakes V_2 or V_1 decelerates.

The vessel V_1 is allowed to travel paralleling with V_2 if the following criterions are satisfied.

- The width of the link is enough for the two vessels to travel parallel, i.e. $W \geq W_1 + W_2$.
- V_1 can overtake V_2 before any of them reach the node A.

The second criterion indicates that vessels are not allowed to overtake any other vessel in any cross region. Otherwise, it will cause more potential conflicts with other vessels around the node A. If V_1 cannot safely overtake V_2, V_1 needs to decelerate by reducing current speed v_1 to the speed v_2.

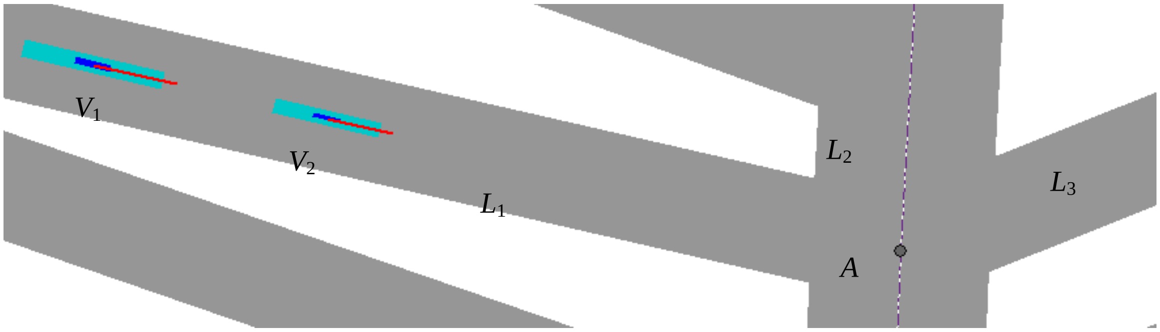


Figure 5.27 Two vessels on a same link.

Table 5.19 Statics for two vessels on a same link.

Vessel	V_1	V_2
Current link	L_1	L_1
Next link	L_2	L_3
Current speed	v_1	v_2
Decelerate rate	r_1	r_2
Max acceleration	a_1	a_2
Domain size	$L_1 \times W_1$	$L_2 \times W_2$
Distance to next node A	D_1	D_2

5.3.4.1 V_1 overtakes V_2

Suppose,

D_{12} : the traveling distance of V_1 after V_1 overtake V_2,

D_{21} : the traveling distance of V_2 after V_1 overtake V_2,

t_{12} : the traveling time of V_1 to overtake V_2.

Then

$$t_{12} = \frac{D_1 - D_2 + L_1 + L_2}{v_1 - v_2},$$

$$D_{12} = t_{12} \cdot v_1,$$

$$D_{21} = t_{12} \cdot v_2.$$

Therefore,

$$D_{12} < D_1 \Rightarrow \frac{D_1 - D_2 + L_1 + L_2}{v_1 - v_2} \cdot v_1 < D_1 \Rightarrow \frac{v_1}{v_2} > \frac{D_1}{D_2 - L_1 - L_2},$$
$$D_{21} < D_2 \Rightarrow \frac{D_1 - D_2 + L_1 + L_2}{v_1 - v_2} \cdot v_2 < D_2 \Rightarrow \frac{v_1}{v_2} > \frac{D_1 + L_1 + L_2}{D_2}.$$

Since $\frac{D_1}{D_2 - L_1 - L_2} > \frac{D_1 + L_1 + L_2}{D_2}$, the second criterion is equivalent to

$$\frac{v_1}{v_2} > \frac{D_1}{D_2 - L_1 - L_2}.$$

5.3.4.2 V_1 decelerates

If V_1 cannot safely overtake V_2, V_1 needs to decelerate by reducing current speed v_1 to the speed v_2. When its speed is reduced to v_2, it only keeps a safety distance to V_2. Thus, the maximum relative movement between the two vessels becomes

$$\tilde{D} = D_1 - D_2 - \frac{L_1 + L_2}{2}. \tag{5.24}$$

If V_1 decelerates at maximum acceleration from current position, the deceleration time is $\left(v_1 - v_2\right)/a_1$, and the relative deceleration distance is

$$D = \frac{\left(v_1 - v_2\right)^2}{2a_1}.$$

Since our system can predict the conflict far before the two vessels meet, it can be assumed that the distance between the two vessels is enough for V_1 to decelerate, i.e.

$$D < \tilde{D}.$$

Therefore, V_1 does not need to decelerate at the maximum acceleration. Instead, V_1 can

- decelerate at a smaller acceleration α_3, or
- keep current speed v_1 for a time period t_1, then decelerate at the maximum deceleration.

In the first case, substituting $\tilde{D} = \frac{\left(v_1 - v_2\right)^2}{2\alpha_3}$ in to Eq.(5.24) gives

$$\alpha_3 = \frac{\left(v_1 - v_2\right)^2}{2\tilde{D}} = \frac{\left(v_1 - v_2\right)^2}{2D_1 - 2D_2 - L_1 - L_2}.$$

In the second case, coupling $\tilde{D} = v_1 t_1 + \frac{\left(v_1 - v_2\right)^2}{2\alpha_1}$ with Eq.(5.24) obtains t_1 as

$$t_1 = \frac{2\alpha_1 \tilde{D} - \left(v_1 - v_2\right)^2}{2\alpha_1 v_1} = \frac{\left(2D_1 - 2D_2 - L_1 - L_2\right)\alpha_1 - \left(v_1 - v_2\right)^2}{2\alpha_1 v_1}.$$

5.3.5 Discussions on delay minimization

As discussed in previous two sections, each vessel has two possible correctives actions to resolve a link/node conflict: changing its trajectory and deceleration. To minimize the total delay, the vessel can choose an action while avoiding a conflict.

Corrective actions would result in delays to vessel operations. Such delays, compared to the delays incurred in the preplanning stage, arise from a dynamic traffic environment which will affect the efficiency of the traffic network. Traffic congestion will appear if delays are accumulated beyond the limitation of tolerable delay in a certain area. Therefore, the decision-making process will check and evaluate that there is sufficient space and time to perform corrective maneuvers while the total delay incurred within the affected area is acceptable.

Two parameters are proposed for minimizing the total delay in the simulation stage: delay tolerance of a link and delay tolerance of a vessel.

Delay tolerance of a link

On each link, different vessels will accumulate different delays. If too many vessels accumulate delays on a same link, traffic congestion is highly possible to appear on this link. Thus, each link is assigned with a delay tolerance. If the accumulated delay on this link reaches the tolerance value, new arrival vessels are not allowed to make turn/decelerate in this link. Instead, if a vessel needs to avoid any conflict on this link, it will have to take actions before it enters this link. According to the conflict prediction, this can be achieved since the conflict can be predicted multi-link-ahead. By default, 2-link-ahead prediction is applied. If a conflict is predicted and the vessel needs to take actions, normally it can take actions after it reaches the third link. However, if delay tolerance is taken into account, it will first check the delay tolerance on the third link. If it is not allowed to take actions on the third link, it will check the second link. If the second link also cannot accept its actions, the vessel will need to take actions in the first link, i.e. the current link.

Delay tolerance for a vessel in the simulation stage

Delay tolerance for each vessel in the preplanning stage refers to slack time which can be absorbed enroute (for arrivals) or at the berths/anchorages (for departures). In the simulation stage, each vessel is assigned with another delay tolerance which represents a maximum acceptable delay for vessel operation within the traffic network. In the simulation system, by default, the vessel with lower priority who acts as the give-way vessel has larger delay tolerance. However, to resolve a conflict, if the vessel reaches its delay tolerance and cannot accept any more delay, the other vessel with higher priority may need to give way to it.

5.3.6 Examples

In this section, some examples are presented to show how to resolve conflicts in preplanning stage and in the simulation stage. In the following examples, the trajectory of each vessel is indicated by a red arrow line. The preplanning time is based on the follow PC configuration

Operation system: Windows 7
CPU: Intel(R) Core(TM) i7-2640M CPU @ 2.80GHz
RAM: 4G

In the first example is set to a section of the Sinki Fairway in the Port of Singapore (Figure 5.28). The schedule chart contains 20 vessels. With the conflict detection method in Chapter 4, seven conflicts are detected in the original schedule chart. The preplanning takes only 1 millisecond. The delay report for the preplanning is presented in Table 5.20. Figure 5.29 shows the meeting situation between the two vessels, vessel_4 and vessel_19, in the region bounded by the blue rectangle in Figure 5.28. After preplanning, the preplanned schedule chart is provided, which still contains one conflict (Figure 5.29(a)). In the simulation stage, vessel_4 can decelerate to avoid the conflict for vessel_19 to keep its court.

Table 5.20 Preplanning report for the example in Figure 5.28.

vessel_19 is delayed for 2882 seconds vessel_14 is delayed for 2543 seconds vessel_16 is delayed for 3525 seconds vessel_5 is delayed for 3010 seconds vessel_4 is delayed for 3905 seconds vessel_15 is delayed for 619 seconds vessel_7 is delayed for 4376 seconds vessel_2 is delayed for 3420 seconds vessel_13 is delayed for 2007 seconds vessel_9 is delayed for 3803 seconds vessel_11 is delayed for 17398 seconds vessel_3 is delayed for 3992 seconds	vessel_1 is not delayed vessel_17 is not delayed vessel_6 is not delayed vessel_18 is not delayed vessel_20 is not delayed vessel_8 is not delayed vessel_12 is not delayed vessel_10 is not delayed
Total delay: 51480 seconds	

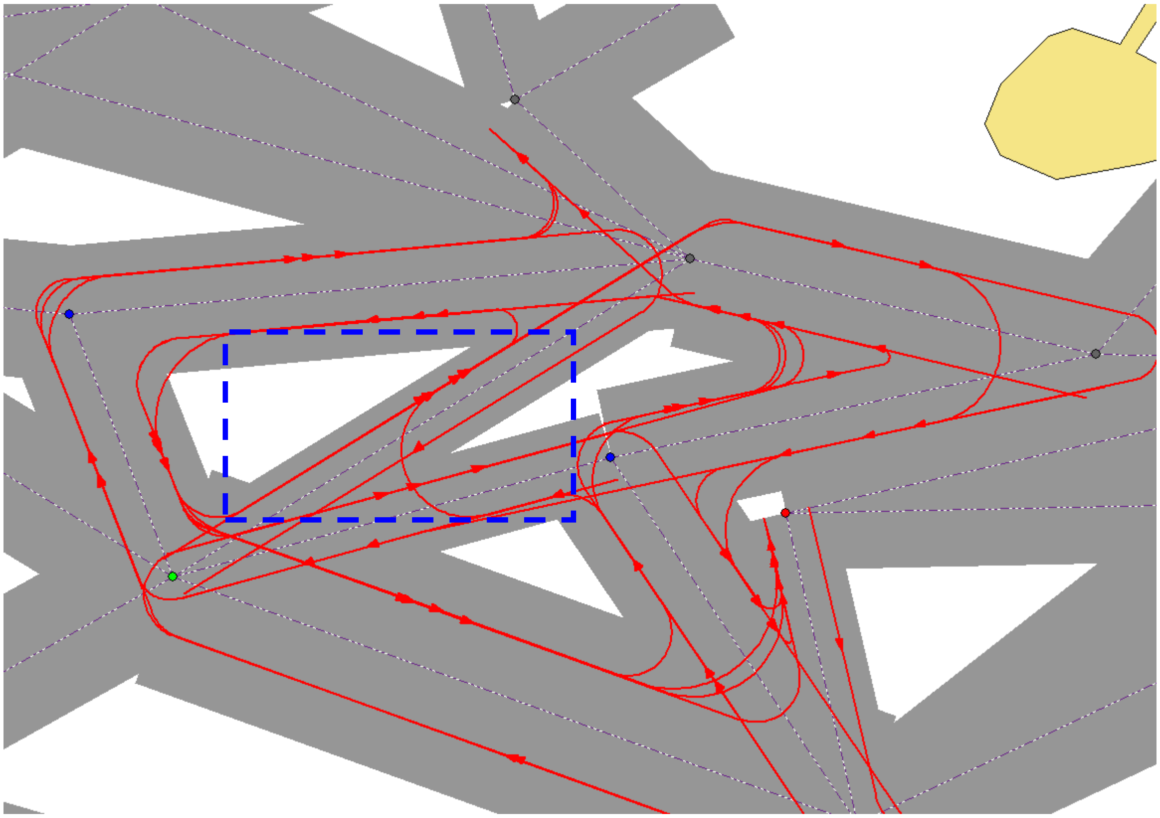
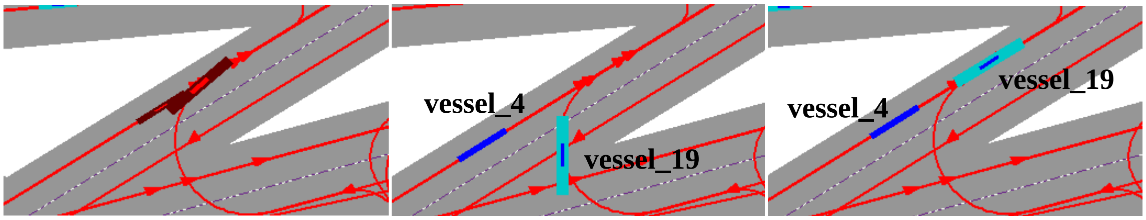


Figure 5.28 A schedule chart with 20 vessels.



(a) (b) (c)

Figure 5.29 vessel_19 and vessel_4.
(a) a conflict between the two vessels in the preplanned schedule chart, (b-c) in simulation stage, vessel_4 decelerates to avoid the conflict.

Figure 5.30 is another example in another sector in the Port of Singapore, a section of the West Keppel Fairway. The original schedule chart contains 33 vessels and involves 21 conflicts. The preplanning takes only 1 millisecond. During the preplanning, 27 vessels have to be delayed. Only 6 vessels with higher priorities keep unchanged (Table 5.21). The region bounded by the blue rectangle is critical. Before the preplanning, most of the conflicts are around this region. Even after preplanning, there are still 3 pairs of vessels conflict with each other. And the three conflicts are all in this region. Figure 5.31 shows the traveling situations in this region during the simulation stage. In Figure 5.31(a), vessel_14, whose priority is lower than vessel_17's priority, gives way to vessel_17. Since vessel_17 comes from its right hand side, it turns right to avoid the conflict. In Figure 5.31(b), vessel_5 comes from the left hand side of vessel_10. Thus, vessel_10 turns left. After making the turn, vessel_14 will travel passing through the back of vessel_17, and vessel_10 will travel passing through the back of vessel_5. According to the simulation, although there are three conflicts after preplanning, after two vessels make turns, all three conflicts are avoided.

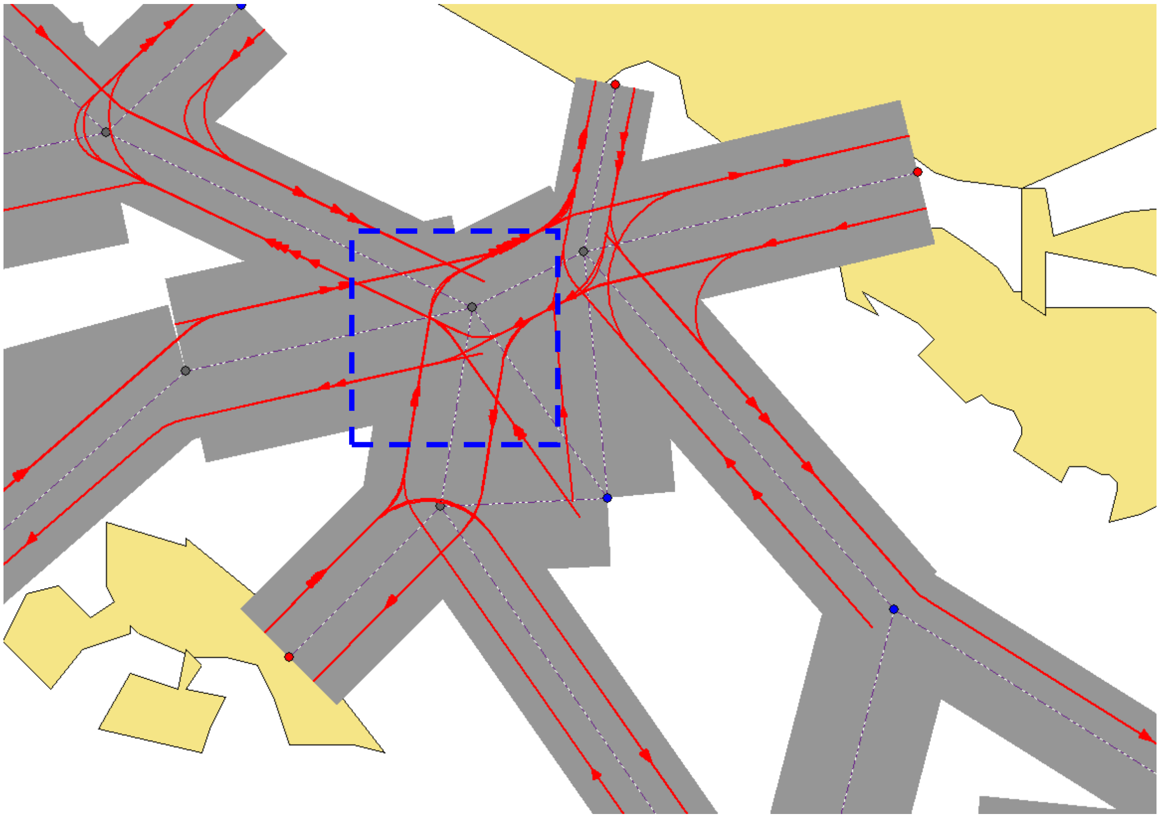
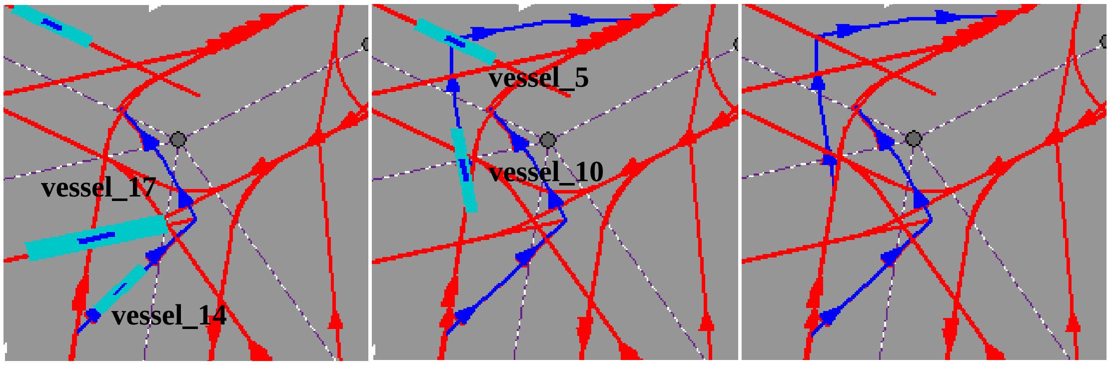


Figure 5.30 A schedule chart with 33 vessels.



(a) (b) (c)

Figure 5.31 Vessels make turns to avoid conflicts.
(a) vessel_14 turn right to give way to vessel_17, (b) vessel_10 turn left to give way to vessel_5, (c) blue lines are turning courses.

Table 5.21 Preplanning report for the example in Figure 5.30.

vessel_30 is delayed for 318 seconds	vessel_29 is not delayed
vessel_3 is delayed for 699 seconds	vessel_13 is not delayed
vessel_9 is delayed for 495 seconds	vessel_25 is not delayed
vessel_6 is delayed for 1153 seconds	vessel_21 is not delayed
vessel_20 is delayed for 780 seconds	vessel_33 is not delayed
vessel_7 is delayed for 811 seconds	vessel_4 is not delayed
vessel_1 is delayed for 2703 seconds	
vessel_31 is delayed for 350 seconds	
vessel_14 is delayed for 1552 seconds	
vessel_28 is delayed for 2703 seconds	
vessel_26 is delayed for 626 seconds	
vessel_23 is delayed for 4105 seconds	
vessel_22 is delayed for 5976 seconds	
vessel_17 is delayed for 4403 seconds	
vessel_32 is delayed for 487 seconds	
vessel_2 is delayed for 765 seconds	
vessel_15 is delayed for 899 seconds	
vessel_10 is delayed for 5573 seconds	
vessel_19 is delayed for 7187 seconds	
vessel_11 is delayed for 3167 seconds	
vessel_27 is delayed for 5514 seconds	
vessel_18 is delayed for 4692 seconds	
vessel_5 is delayed for 7805 seconds	
vessel_8 is delayed for 5737 seconds	
vessel_12 is delayed for 7738 seconds	
vessel_24 is delayed for 436 seconds	
vessel_16 is delayed for 5368 seconds	
Total delay: 82042 seconds	

5.4 Summary

This chapter presented a two stages conflict resolution strategy for the simulation: the preplanning stage and the simulation stage. In the first stage, preplanning is adopted to remove most of the conflicts in the original schedule chart. In the simulation stage, vessels' actions, deceleration and turning, to avoid the conflicts are simulated. Delay calculation is enabled in both stages to evaluate the simulation. Vessel's priority acts as an important factor in the simulation. Vessels with higher priority can have smaller delay. The preplanning can be performed very fast, and acts as a preprocessing for the simulation stage. Without preplanning, there may be too many conflicts in a local region thus may cause traffic jam in the region. Many examples are presented to show the efficiency of the combination of the preplanning and the simulation. All potential conflicts can be detected and resolved.

Chapter 6

Marine traffic conflict simulation system

Chapter 4 introduces how to detect conflicts, and Chapter 5 presents how to resolve conflicts. Marine traffic conflict simulation system is developed as an experimental platform for demonstrating and evaluating the methods addressed in Chapters 4 and 5. This chapter will introduce the overall design of the simulation system. Five functional modules of the system will be illustrated in detail. An application of the model will be demonstrated using the Port of Singapore as an example. Traffic statistics are collected according to the input data from the Port of Singapore. Model demonstration includes a series of simulation trials based on the statistic data. Sensitivity analysis will be performed to evaluate the effect of variations in model inputs. Traffic volume as influence factor will be used for evaluating the impact on conflict situations and the total delays. Expected result is a reasonable change of conflicts when traffic volume is changed, in regards with the number of conflicts and total delays resulted from conflict resolution measures.

6.1 Introduction

A port traffic system is a random, dynamic and complex system which contains a large number of interfering variables. Simulation is an attempt to model a real life system on a computer so that it can be studied to see how the system works. By changing variables, predictions about the behaviors of the system may be made. The development of a marine traffic conflict simulation system has the following advantages:

- It can access incident/accident scenes without real participation;
- It can reduplicate a specific traffic scenario for testing different conflict resolutions;
- It can modify system operation conditions so as to predict traffic behaviors;
- It can deal with complicated conflict situations;
- It can simulate random vessel movements.

This study aims to develop a simulation system, called “Marine Traffic Conflict Simulation System”, which can be run on advanced microcomputers or graphic workstations. Algorithms for conflict prediction and conflict resolution outlined will be implemented through the simulation platform. Moreover, the inclusion of the simulation system in this research is to provide a platform/basis for extending the work into real-time conflict resolution, to point out possible directions of further work.

6.2 Implementation of the simulation system

6.2.1 Development environment

The simulation system is developed under Windows XP environment, and it is designed to have enough capacity to process a large amount of data. Hardware and software requirements are set as follows:

1) Simulation terminals: advanced microcomputer and a graphic workstation;
2) Microsoft Visual C++.

6.2.2 System overview

Figure 3.1 is the framework of the simulation system. It consists of 5 modules: the simulation control module, the graphic display module, the data processing module, the conflict detection module, and the conflict resolution module. The following sections will detail the functions of these modules.

6.2.3 Graphic interface and functional modules

The simulation system is expected to predict conflicts in a marine traffic system, and to provide feasible conflict solutions while minimizing delay resulting from anti-conflict measures/actions. It is expected to work within the whole seaport or within a critical region in the seaport. Functions of the system are summarized below:

1) Loads map for the sea space and draws fairways within the seaport;
2) Displays vessel movements in any time period with given speeds, origins and destinations;
3) Detects possible conflicts and predicts potential conflict risk;
4) Provides actions/measures to avoid conflict and calculates the associated delay;
5) Provides optimized rules for conflict avoidance so as to minimize total delay in the traffic network;
6) Provides functions of screen capture, scene replay, data record and print;
7) Processes the input data to obtain the arrival information at each node;
8) Analyzes the changes of traffic system by changing system parameters.

With the above considerations, five modules are designed in the simulation system. The functions of each module will be described. The simulation system is developed with a friendly graphical user interface (GUI). Under this GUI, simulation details are displayed by dynamic graphics; the simulation outcomes of the system are presented in simple and vivid graphics format; and the simulation process is monitored and intervened through human-computer interaction. Main interface of the simulation model is shown in Figure 6.1. On the left hand side is the control panel. The Port of Singapore is shown on the right hand side.

The five basic modules (Figure 6.2) are controlled from the control panel, including

- graphic display (Figure 6.2(a)),
- data processing (Figure 6.2(b)),
- simulation control (Figure 6.2(c)),
- conflict detection (Figure 6.2(d)), and
- conflict resolution (Figure 6.2(e, f)).

The following sections will detail these modules.

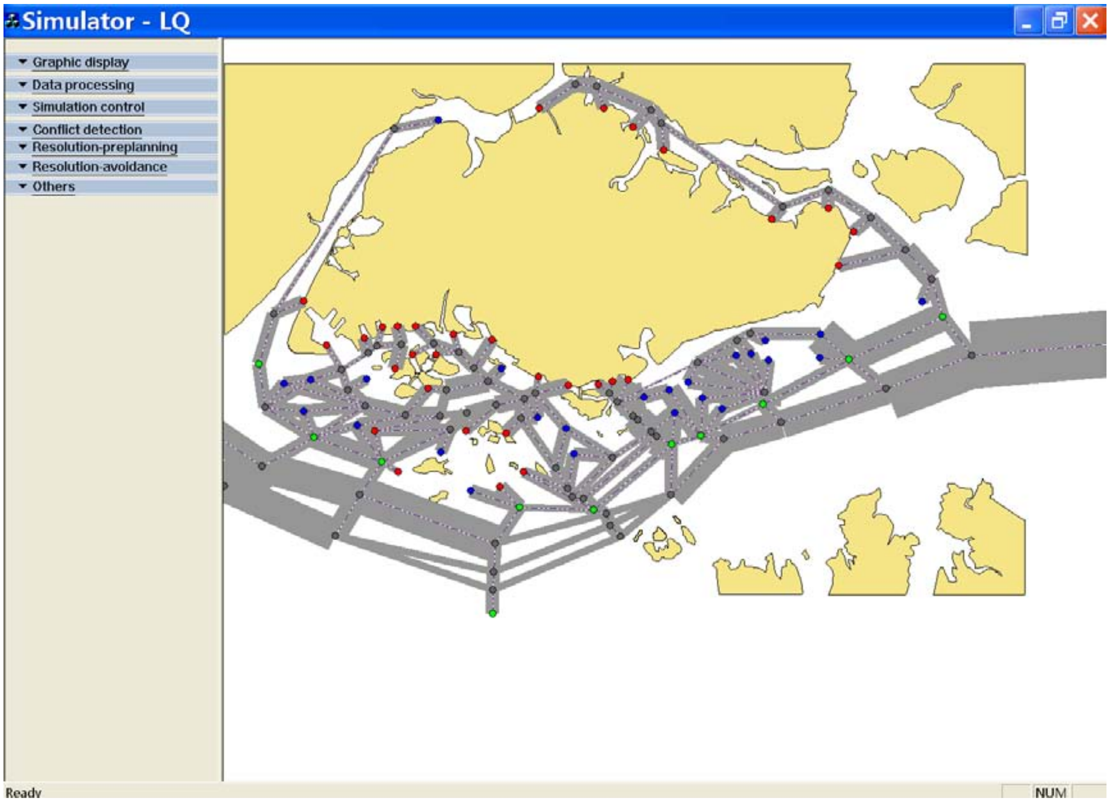


Figure 6.1 The user interface of the simulation.

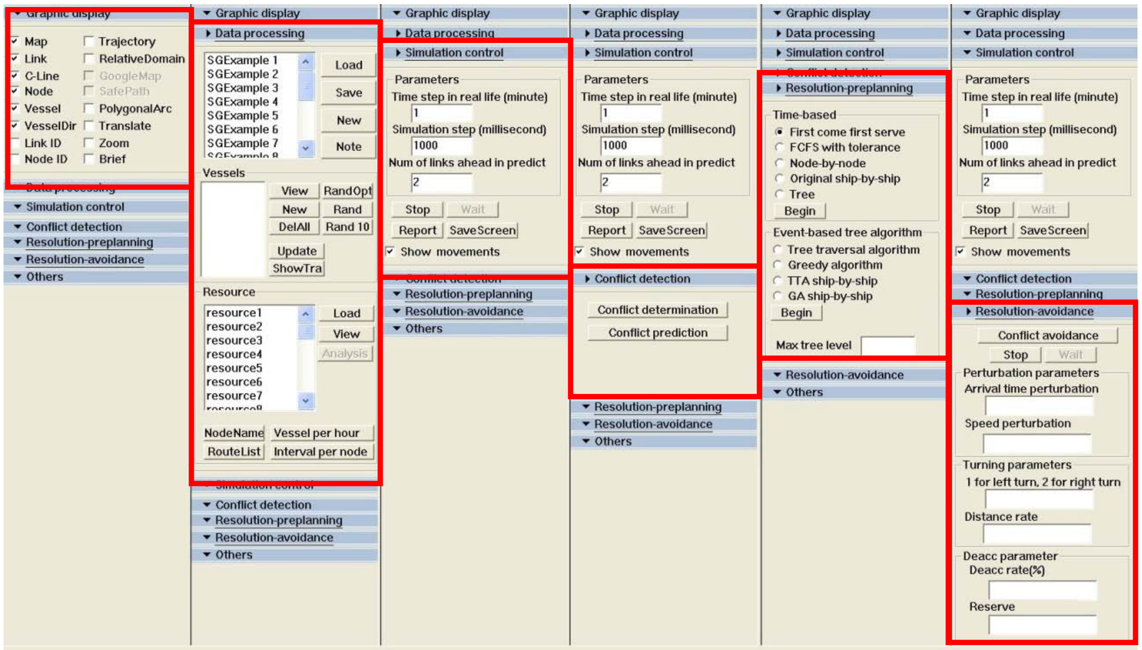
6.2.3.1 Graphic display module

This module provides visual representation of dynamic graphic simulation. It also serves as the human-computer interface.

Seaport visualization

This module controls the visibility of different seaport components. As shown in Figure 6.2(a), users can simply check a component to visualize it or uncheck the component to hide it.

- Map,
- Links and their IDs,
- Nodes and their IDs,
- Vessels and their directions, and
- The vessels' trajectories.



Time step in real life (minute): 1
Simulation step (millisecond): 1000
Num of links ahead in predict: 2
☑ Show movements
● First come first serve

(a) (b) (c) (d) (e) (f)

Figure 6.2 Five modules for simulation.
(a) graphic display, (b) data processing, (c) simulation control, (d) conflict detection, (e) preplanning, and (f) avoidance.

Region selection

Users can use mouse to zoom in/out the map to view a local region from the seaport (Figure 6.3). This enables users to observe vessel movements and traffic situations in certain regions which can be identified by users.

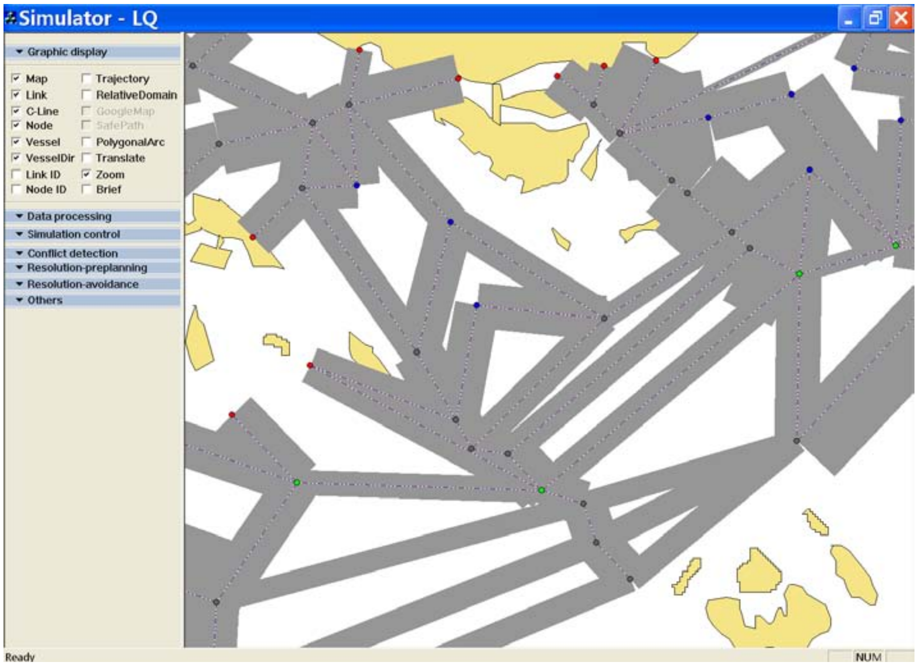
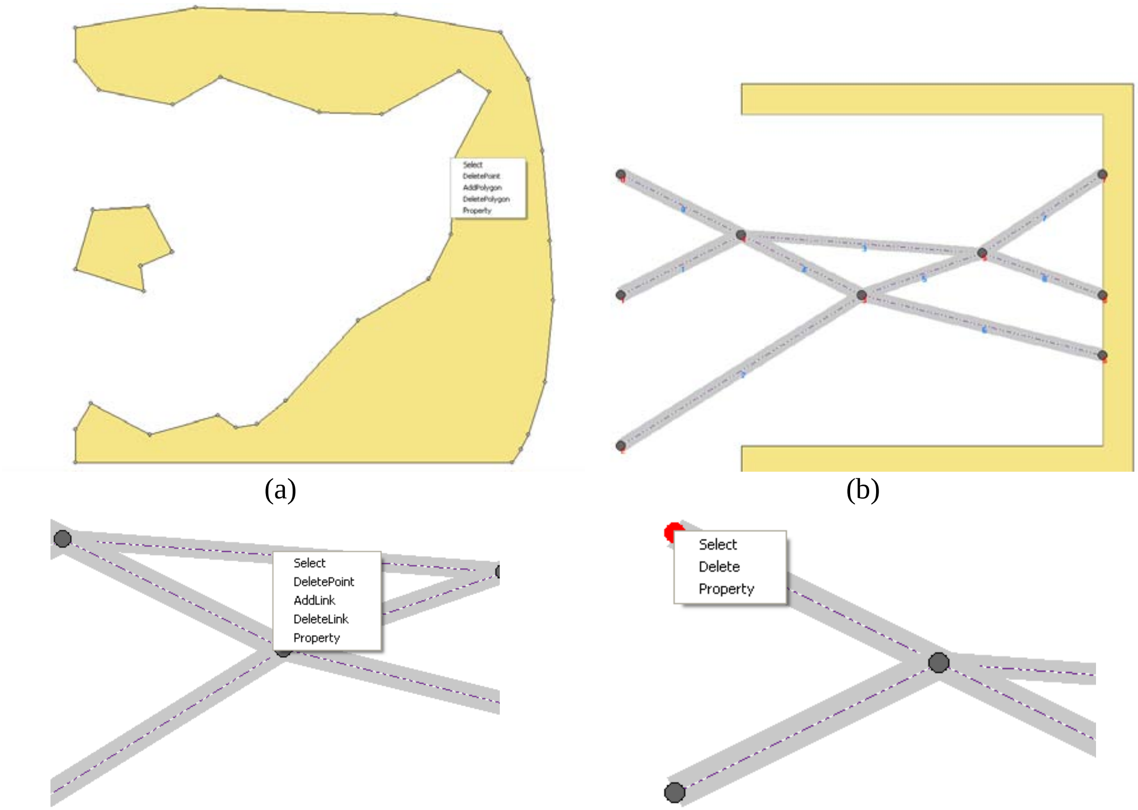


Figure 6.3 Zoom in/out of the map.

Interactive operations

Some time users need to construct a virtual environment for estimating conflict situation or evaluating algorithms. Thus, the simulation system provides interactive operations for setting background map. By mouse dragging, users can draw map for the sea space and the land in its vicinity, and revise the boundary of the sea space (see Figure 6.4(a)). Similarly, interactive operations are available for traffic network construction: build traffic network using mouse (see Figure 6.4(b)). Users can add/ delete nodes and links, as well as change the property of them, e.g. change the width of a link (see Figure 6.4(c, d)).



(a) (b)

(c) (d)

Figure 6.4 Interactive operations.
(a) map editing, (b) net construction, (c) link editing, and (d) node editing.

6.2.3.2 Data processing module

Data processing module contains three sub-modules

- Schedule chart sub-module,
- Vessel control sub-module, and
- Resource sub-module.

Schedule chart

The schedule chart sub-module provides the basic functions to edit the schedule chart. After loading a predefined schedule chart, users can view/edit/delete existing schedules, add new schedules.

Vessel control

Each vessel in the seaport travels according to a schedule. Its route is defined by a sequence of nodes that it will pass one by one. Its trajectory is controlled and updated according to its route and its rate of turning (ROT). In Figure 6.5, the numbers in blue are IDs for links, the numbers in red are IDs for nodes. In the system, each route is represented by a sequence of links' IDs, and users can edit the route of a vessel to change its trajectories.

Basically, a vessel in the seaport contains the following information

- Vessel name,
- Vessel type,
- Vessel speed,
- LOA,
- Veam,
- Route.

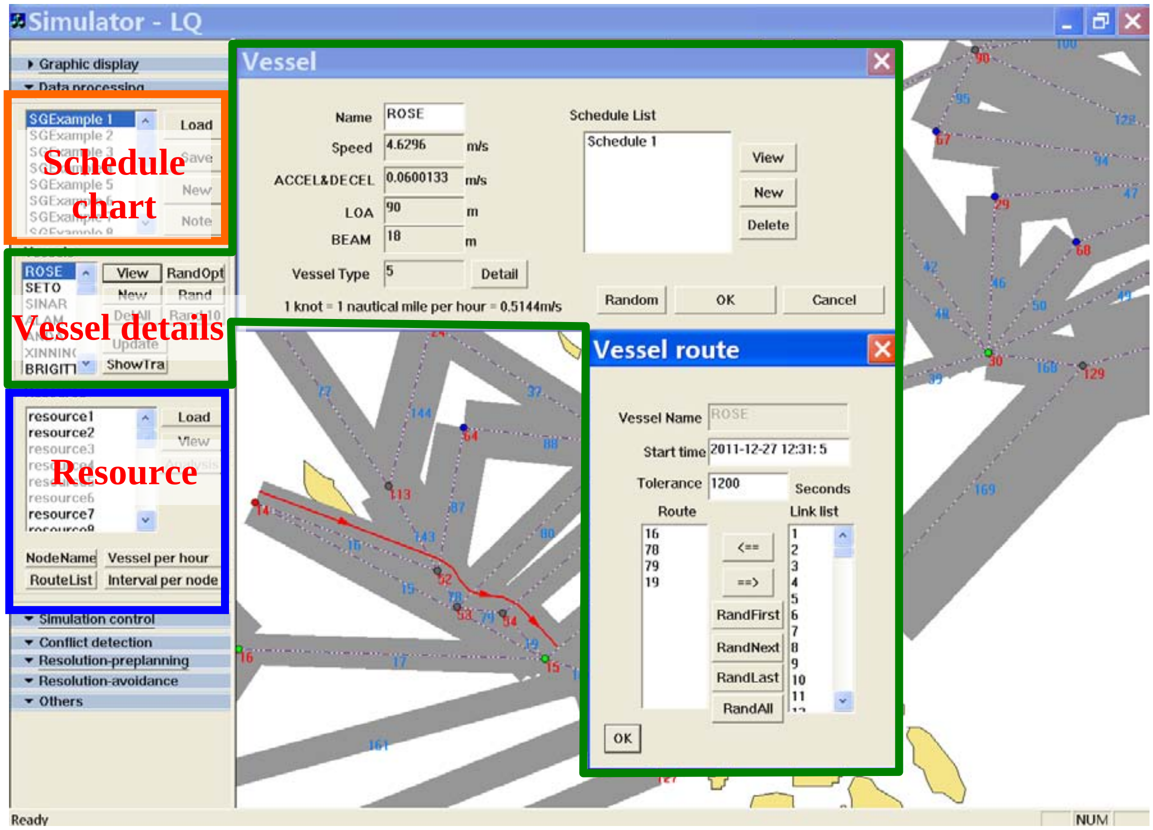
Resources

The resource sub-module loads and analyzes the input data. It is to read and turn the input data into the format used in the simulation system (will be detailed in Section 6.3.2). The input data includes

- Traffic network of the seaport of Singapore,
- Different types of vessels,
- The predefined routes from one node to another, and
- The schedule chart of each day during April 1998.

The input data includes 901 predefined routes, each of which tells how to make a route from one node to another node (Figure 6.6). For example, the route from NODE16 to NODE19 is the link list LINK16-78-79-19 (the red arrow curve in Figure 6.5).

The schedule charts of Singapore are analyzed in this module. The statistics for vessel arrivals are presented in Appendix C. Also, the average arrival time interval at each node can be calculated from the input data.



SGExample 1
SGExample 2
SGExample 3
SGExample 4
SGExample 5
SGExample 6
SGExample 7
SGExample 8

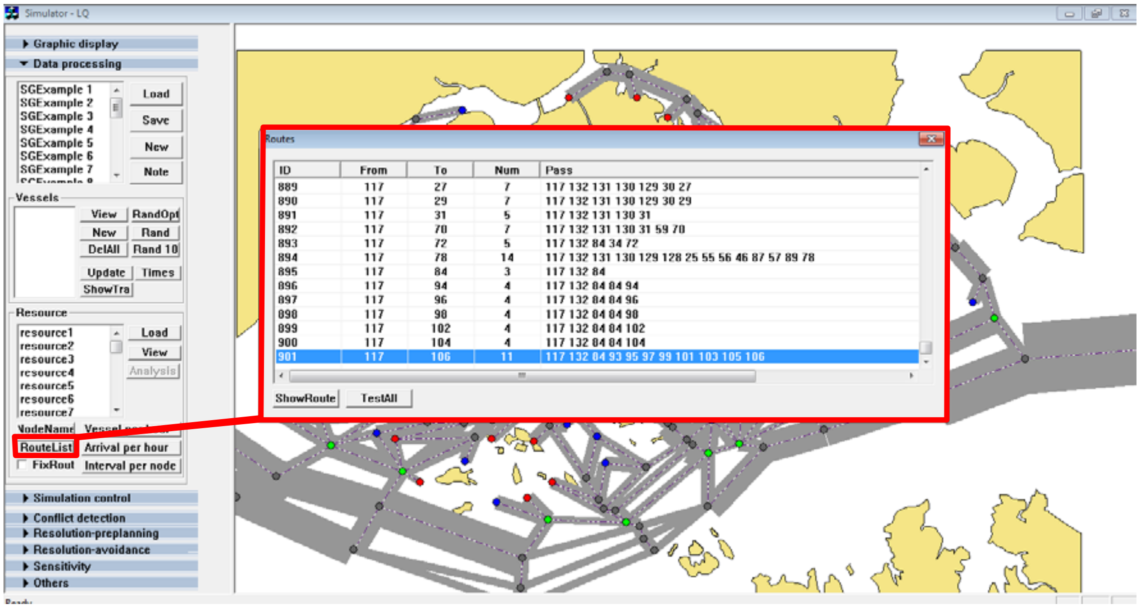
ROSE
SETO
SINAR
XINNING
BRIGITT

resource1
resource2
resource3
resource4
resource5
resource6
resource7
resource8

Name: ROSE
Speed: 4.6296 m/s
ACCEL&DECEL: 0.0600133 m/s
LOA: 90 m
BEAM: 18 m
Vessel Type: 5
1 knot = 1 nautical mile per hour = 0.5144m/s
Schedule List: Schedule 1

Vessel Name: ROSE
Start time: 2011-12-27 12:31:5
Tolerance: 1200 Seconds
Route: 16 78 79 19
Link list: 1 2 3 4 5 6 7 8 9 10 11

Figure 6.5: Data processing module.



SGExample 1
SGExample 2
SGExample 3
SGExample 4
SGExample 5
SGExample 6
SGExample 7
SGExample 8

resource1
resource2
resource3
resource4
resource5
resource6
resource7

ID	From	To	Num	Pass
889	117	27	7	117 132 131 130 129 30 27
890	117	29	7	117 132 131 130 129 30 29
891	117	31	5	117 132 131 130 31
892	117	70	7	117 132 131 130 31 59 70
893	117	72	5	117 132 84 34 72
894	117	78	14	117 132 131 130 129 128 25 55 56 46 87 57 89 78
895	117	84	3	117 132 84
896	117	94	4	117 132 84 84 94
897	117	96	4	117 132 84 84 96
898	117	98	4	117 132 84 84 98
899	117	102	4	117 132 84 84 102
900	117	104	4	117 132 84 84 104
901	117	106	11	117 132 84 93 95 97 99 101 103 105 106

Figure 6.6: The list of routes.

6.2.3.3 Simulation control module

Simulation control module controls the simulation running:

- Time step in real life: s minutes
- Time step in simulation: t milliseconds
- Number of links ahead in conflict prediction: n.

The basic operations such as “run”, “pause”, “continue”, “stop” are also provided in this module. The time step in the simulation, t milliseconds, corresponds to the time step in real life, s minutes. For a schedule chart of 24-hours, the running time of a simulation is

$$24*60/s*t.$$

The total running time for a simulation is depending on the values of s and t. To reduce the total running time, it only needs to increase s and reduce t. However, s cannot be too big. Otherwise, a potential conflict may be omitted. The value for s is to guarantee that all conflict can be detected. The default values for s and t are taken as

$$s = \min_i \{\frac{l_i}{v_i}\}, \qquad t = 100.$$

where l_i and v_i are the length and the speed of the i-th vessel.

6.2.3.4 Conflict detection module

Conflict detection is consisting of two parts

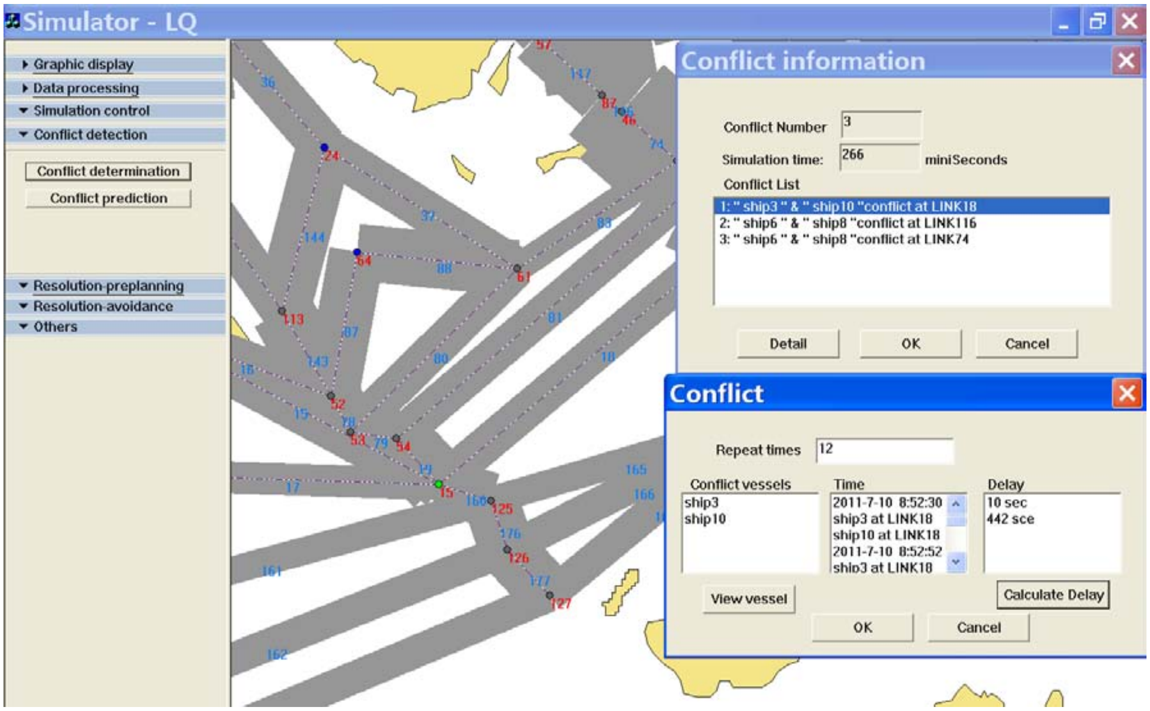
- Conflict determination presented in Section 4.2.1, which uses the vessels’ domains to detect the conflicts in a schedule chart,
- Conflict prediction presented in Section 4.2.2, which uses relative movement of the domains to predict the conflict.

The output of conflict detection is a report showing all the details for the conflicts within a schedule (Figure 6.7), including

- the two vessels in a conflict,
- the time when a conflict is detected,
- the delay required for each vessel to avoid this conflict.

6.2.3.5 Conflict resolution module

As introduced in Chapter 5, conflict resolution module contains two parts: the preplanning stage (Figure 6.8) and the simulation stage (Figure 6.9).



Conflict information

Conflict Number 3

Simulation time: 266 miniSeconds

Conflict List

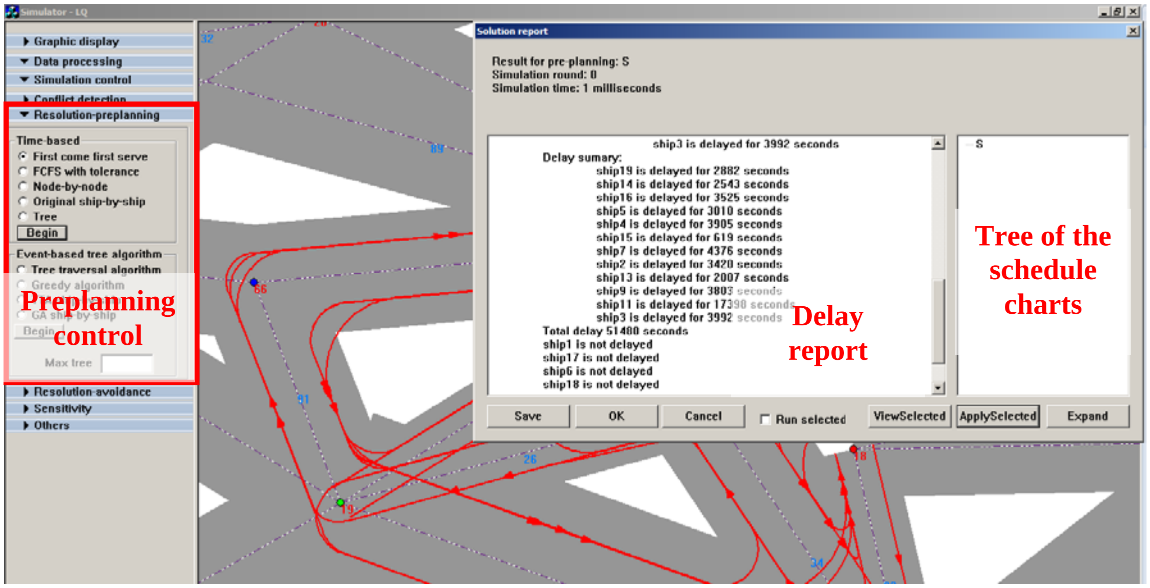
1: " ship3 " & " ship10 "conflict at LINK18
2: " ship6 " & " ship8 "conflict at LINK116
3: " ship6 " & " ship8 "conflict at LINK74

Conflict

Repeat times 12

Conflict vessels	Time	Delay
ship3 ship10	2011-7-10 8:52:30 ship3 at LINK18 ship10 at LINK18 2011-7-10 8:52:52 ship3 at LINK18	10 sec 442 sce

Figure 6.7: Details for each conflict.



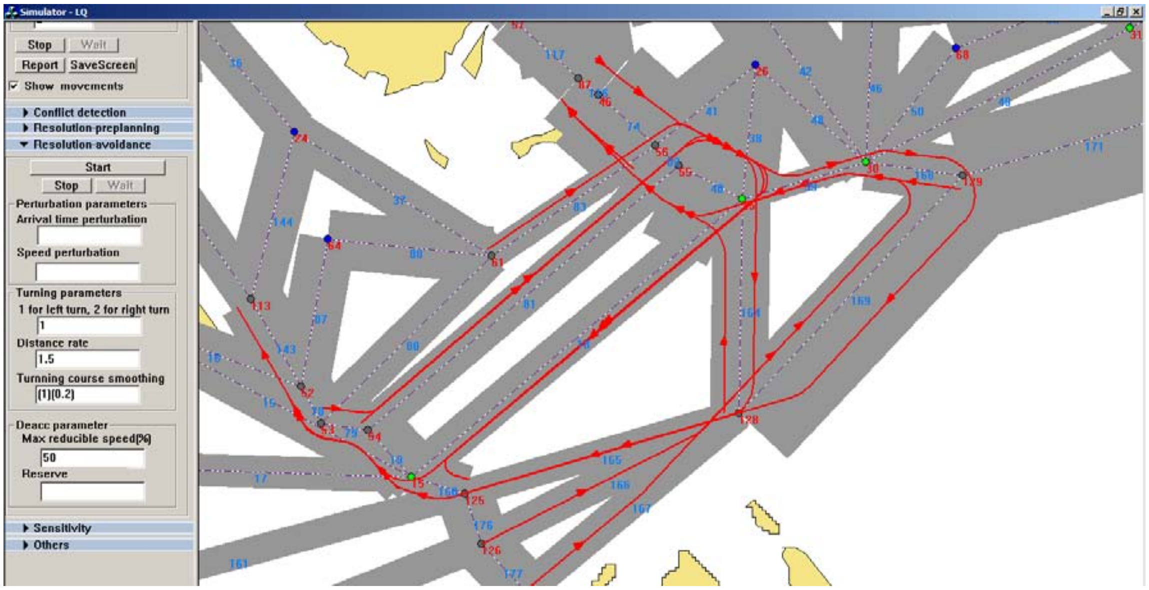
Solution report

Result for pre-planning: S
Simulation round: 0
Simulation time: 1 milliseconds

```
                    ship3 is delayed for 3992 seconds
    Delay sumary:
            ship19 is delayed for 2882 seconds
            ship14 is delayed for 2543 seconds
            ship16 is delayed for 3525 seconds
            ship5 is delayed for 3010 seconds
            ship4 is delayed for 3905 seconds
            ship15 is delayed for 619 seconds
            ship7 is delayed for 4376 seconds
            ship2 is delayed for 3420 seconds
            ship13 is delayed for 2007 seconds
            ship9 is delayed for 3803 seconds
            ship11 is delayed for 17390 seconds
            ship3 is delayed for 3992 seconds
    Total delay 51480 seconds
    ship1 is not delayed
    ship17 is not delayed
    ship6 is not delayed
    ship18 is not delayed
```

Resolution-preplanning: Time-based: ● First come first serve

Figure 6.8 Conflict resolution module: preplanning.



☑ Show movements

Perturbation parameters
Arrival time perturbation:
Speed perturbation:

Turning parameters
1 for left turn, 2 for right turn: 1
Distance rate: 1.5
Turnning course smoothing: (1)(0.2)

Deacc parameter
Max reducible speed(%): 50
Reserve:

Figure 6.9 Conflict resolution module: corrective measures.

6.3 Model demonstration

A general framework has been developed for the simulation system. An application model of the simulation system will be demonstrated using the seaport of Singapore as an example (see Figure 6.10). The traffic network of the Port of Singapore is shown in Figure 4.2.

Model validation has not been done at current stage. It is noticed that model validation requires a substantial amount of sound data from real world. In this research, the input data used in model demonstration was collected 30 days' traffic data in April 1998. There is difficulty in accessing recent data maintained by port authorities, because quite a lot of the seaport information needed is confidential and sensitive and are protected.

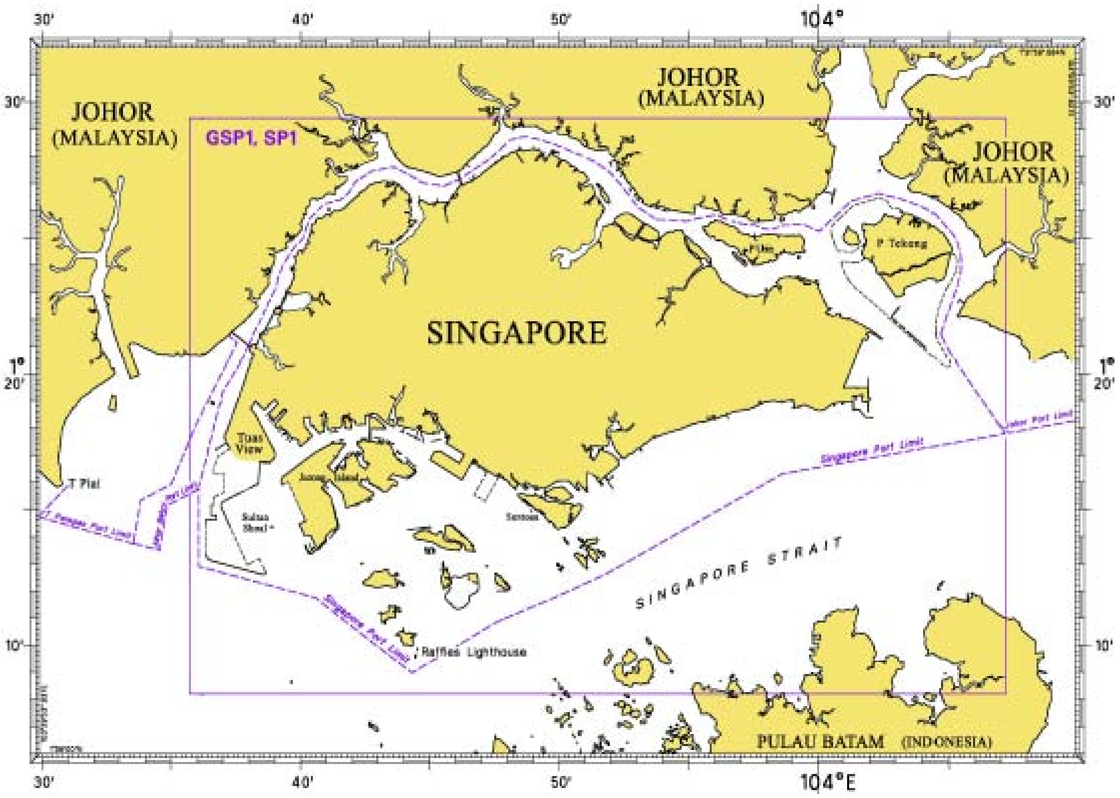


Figure 6.10 The Singapore Chart from MPA.

In this section, target of model demonstration is to verify that the schedule chart provided by the preplanning is feasible, where almost all potential conflicts under original vessel schedules are eliminated. Simulation is also used to demonstrate that, although random conflicts will occur in a dynamic traffic environment, such conflicts can be resolved by proper corrective measures. Therefore, illustrating model rigor and applicability is more important than having a recent dataset for research purpose. In other words, the scientific originality and contribution remains.

6.3.1 Assumptions

The simulation experiments which will be presented in following sections are set up upon certain assumptions,

- The simulation system is a simplification not the same as real life situations. For example, a fairway is simplified as a set of links; a node is used to represent an area of in the seaport; vessels are restricted to travel following a straight line along a link.
- Traffic rules and regulations in the Port of Singapore are not taken into account.
- Geographic conditions and weather, e.g. the effect of wind and tide are not considered.

A simulation model requires a substantial amount of input data on vessel speeds, and dimensions of clearance areas. These data are not completely acquired in this research. In such a case, simplified assumptions and approximations are made to estimate the data required. After the model has been built, it is important to assess whether the simulation outcomes are correct and accurate.

6.3.2 Input data processing

Input data used for model demonstration include data on fairways, vessel dimensions, vessels' domains, vessel speeds, vessel types and navigational schedules. These data are extracted from survey records containing vessel movements and their characteristics. However, information in survey records may be miscellaneous stored, e.g. a larger number of vessels with various shape, dimensions and operation characteristics. And, some information in the survey record may not be used directly by the simulation system, e.g. irregular curves in fairways. Thus, it is necessary to process survey record before inputting them into the simulation system. All proceeded data are stored in the format of MS-Excel files, so that the simulation system can easily extract information from the files.

6.3.2.1 Traffic network

The traffic network is consisting of nodes and links. Generally, a node corresponds to an area in the seaport for specified functions, and a link is a section of a fairway. These data are built according to the information obtained from the Maritime and Port Authority of Singapore (MPA).

There are 132 nodes covering four area types shown in Table 6.1. Take the boarding point for example. MPA has defined 4 pilotage districts. Every vessel navigating in any pilotage district should be under pilotage, i.e. vessels will have seaport staffs on board and guide the navigations. On the boarders of these pilotage districts, MPA has setup 6 pilot boarding grounds (MPA 1998)

- Eastern Boarding Ground "A" (PEBGA)
- Eastern Boarding Ground "B" (PEBGB)
- Southern Boarding Ground (PSBG)
- Western Boarding Ground “A” (PWBGA)
- Western Boarding Ground “B” (PWBGB)
- East Johor Strait Boarding Ground (PJSB)

The input data contains 9 boarding points, all of which are locating inside the 6 boarding grounds. As from Figure 4.2, all vessels enter/leave the Port of Singapore need to pass a boarding point.

Each node has its coordinates. For anchorage areas, berthing areas, and boarding points, each node contains several regions with different names. For example, NODE10 is a berthing area, which is divided into 9 working regions with names:

OE1, OE2, OE3, OE4, OE5, OE6, OE, OEJT, OESHP.

Table B.1 in Appendix B lists all the 69 nodes with its number of regions. Appendix B also lists all the nodes used in the simulation.

Table 6.1 Statistics of nodes.

Type of nodes	Number of the nodes	Correspondence to Figure 4.2
Anchorage areas	27	Blue dots
Berthing areas	33	Red dots
Boarding points	9	Green dots
Separation points and intersections	63	Black dots

The Port of Singapore contains a set of fairways and channels (MPA 2012) (see Figure B.2 in Appendix B), including

- West Jurong Channel,
- East Jurong Channel,
- Temaske Fairway,
- Sinki Fairway,
- West Keppel Fairway,
- East Keppel Fairway,
- Jong Fairway,
- Sisters Fairway,
- Southern Fairway, and
- Eastern Fairway.

In the simulation system, each fairway/channel is divided into links. The input data include 180 links covering all the fairways/channels (Figure 6.11). These links are separated into two groups: single-way links and dual-way links (Table 6.2). Graphically, a dual-way link contains a dash line separating the lanes (Figure 6.12(a)). In real life situation, this dash line usually acts as buoys forbidding vessels to cross. In the system, a vessel can travel on the left-hand-side or right-hand-side of the link. By default, it takes the left-hand-side, i.e. a vessel traveling from NODE131 to NODE132 will always have the buoys on its right hand side. Thus, the fairway from NODE132 to NODE131 is wider than the fairway from NODE131 to NODE132. In a single-way link, such as LINK73 in Figure 6.12(b)), a vessel can only travel from NODE49 to NODE45. If a vessel wants to travel passing NODE45 and NODE49, it cannot cross LINK73. Appendix B also provides all the links used in our system.

Table 6.2 Statistics of links.

Type of links	Number of the links	Correspondence to Figure 4.2
Single-lane	16	Rectangular regions with central lines
Dual-lane	164	Rectangular regions without central lines

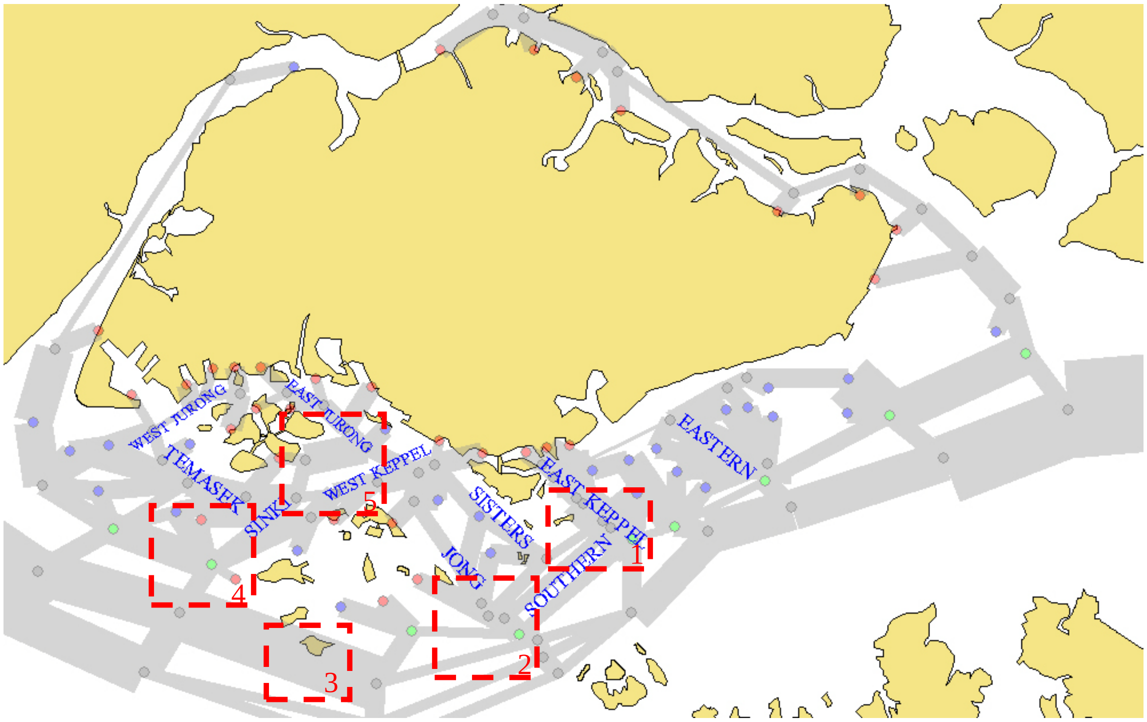
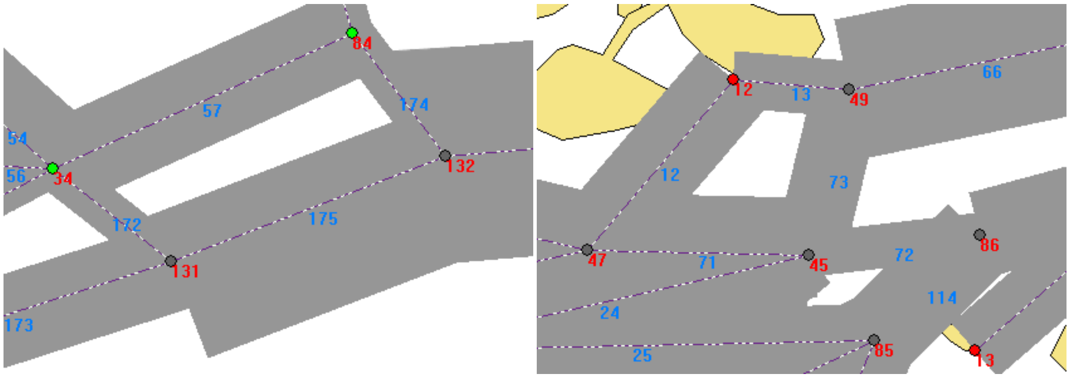


Figure 6.11 Links and critical regions.



(a) (b)

Figure 6.12 Two types of links.
(a) a dual-lane link, LINK175, between NODE131 and Node132, and (b) a single-lane link, LINK73, between Node49 and NODE45.

6.3.2.2 Vessels

Vessels are consolidated into groups according to the nature of their usage, size, speed and operation characteristics. The simulation system adopts 19 types of vessels (Table 6.3) with 7 different types of speeds (Table 6.4). Irregular data about vessel shape, vessel’s domain and fairways are preprocessed to correspond to assumptions made in the mathematical model. In our simulation system, the speed of the vessel changes from time to time, i.e. deceleration, acceleration and turning (Table 6.4). The shape of the domain for each vessel also keeps change according to its speed (Table 6.3).

Table 6.3 List of vessel types.

Type	Domain@ < 6 knots[1]		Domain@ 6-12 knots[1]		Domain@ > 12 knots[1]		swing time[2]	speed type
	Length[3]	Width[3]	Length[3]	Width[3]	Length[3]	Width[3]		
1	3L	3B	3.5L	3B	4L	3B	15	2
2	3L	3B	3.5L	3B	4L	3B	15	2
3	3L	3B	3.5L	3B	4L	3B	15	2
4	3.5L	3B	4L	3B	4L	3B	25	5
5	3.5L	3B	3.5L	3B	4L	3B	25	2
6	3.5L	3B	3.5L	3B	4L	3B	30	5
7	4L	3B	4L	3B	4.5L	3B	35	6
8	4L	4B	4L	3B	4.5L	3B	45	7
9	3L	3B	3.5L	3B	4L	3B	25	3
10	3L	3B	3.5L	3B	4L	3B	30	4
11	3L	3.5B	3.5L	3.5B	4L	3.5B	40	4
12	3.5L	3B	4L	3B	4.5L	3B	15	8
13	3.5L	3B	4L	3B	4.5L	3B	30	8
14	3L	3B	3.5L	3B	4L	3B	15	2
15	3L	3B	3.5L	3B	4L	3B	25	4
16	2.5L	3B	3L	3B	3.5L	3B	10	1
17	2.5L	3B	3L	3B	3.5L	3B	15	1
18	2.5L	3B	3L	3B	3.5L	3B	25	4
19	3L	3B	3.5L	3B	4L	3B	15	2

[1] 1 knot = 1 nautical mile per hour = 0.5144m/s.
[2] the unit is minute.
[3] L for LOA, B for beam.

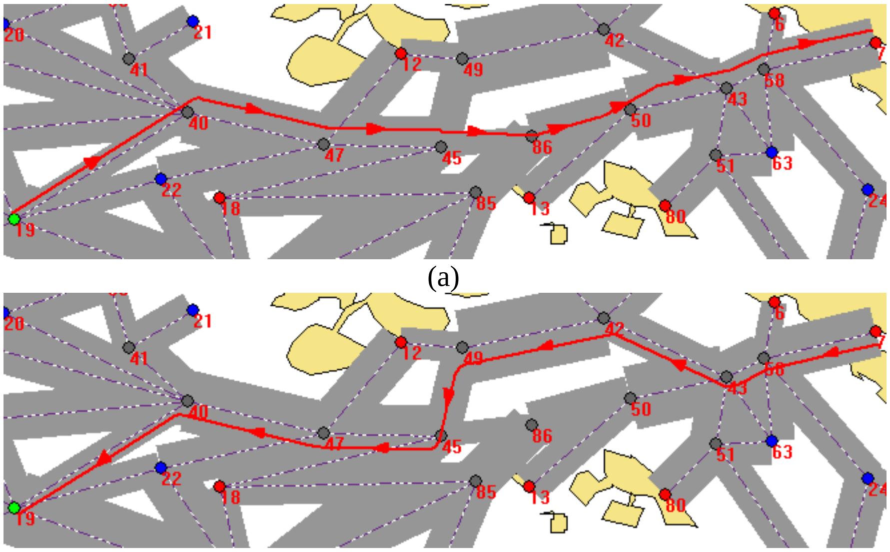
Table 6.4 List of speed types.

Description	acceleration[1]	speed	bend speed	strait speed
Tugs, Ferries, Coasters	6	10	10	10
Bunker vessels, coaster & freighters	7	9	8	12
Feeder conships	9	12	6	18
3 generation container	12	12	7	18
Tanker-medium	14	10	8	12
Tanker-large	18	8	6	12
VLCC	22	8	6	12
Tow boats, barges	15	8	6	8

[1] The unit is knots/minute.

6.3.2.3 Routes

For a vessel travelling from one node to another, the route is pre-defined. As shown in Figure 6.6, the input route data include 901 pre-defined routes. In general, the first and the last nodes of each route are anchorage nodes, berthing nodes, or boarding nodes. Each route only can pass intersecting nodes. For example, in Figure 6.13(a), the route starts from NODE19 and ends at NODE7. NODE19 is a boarding node, and NODE7 is a berthing node. The route passes NODE40, NODE47, NODE45, NODE86, NODE50, NODE43, and NODE58, which are intersecting nodes. One the other hand, the route in Figure 6.13(b) does not pass NODE50 and NODE86. Instead, it passes NODE42 and NODE49. Moreover, each route does not pass any anchorage node, berthing node, or boarding node.



(a)

(b)

Figure 6.13 The routes between two nodes.
(a) the route from NODE19 to NODE7, and (b) the route from NODE7 to NODE19.

6.3.2.4 Schedules

The schedule data for 30 days in the Port of Singapore are collected for processing. Statistics data are presented in Appendix C. Since only a few schedules are collected on 16-April, the data of 16-April is skipped from the following calculation. In summary, there are 17751 schedules. Table 6.5 lists four schedules from the input. For each schedule, the system needs to derive the vessel, the route, and the arrival time. The arrival time can be directly obtained from the input. Take the vessel ‘ENNERDALE’ as an example to see how to extract the vessel and the route.

To extract the vessel, the system needs first obtain the vessel type by its type code and the LOA from Table 6.6. The vessel type for ‘ENNERDALE’ is 5. Then, the system can use the vessel type to derive the vessel’s navigation information from Table 6.3, including the speed, the domain, and the acceleration. During the simulation, the vessel's domain is changed according to different navigation speeds.

The route is defined by the node it comes from and the node it goes to. 'ENNERDALE' comes from 'ALGAS', which is the anchorage point NODE66, and goes to 'OSPS1', which is the berthing point NODE11. Thus, the route for 'ENNERDALE' is the route from NODE66 to NODE11.

Table 6.5 Four schedules from input.

Vessel name	Type Code	LOA	From	To	Arrival time
ENNERDALE	LP	100	ALGAS	OSPS1	9804010000
ORION TRADER	TA	332	SEAE	SEAW	9804010209
ADOPHIUS	TA	65	AESPC	ASSP	9804010234
LIMIA 1	BA	46	SEAE	RJR11	9804010400

Table 6.6 Vessel type by LOA and the type code.

Type	LOA range	Type code
1	0 to 30 m	FV,SV,TU,WA,WB,OT
2	31 to 90 m	CF,CH,CO,DL,FR,LC,OT,PT,RV,TU,WA,WB,FV,SV,SA
3	91 to 110 m	CF,CO,FV,OT,FR,LC,CH,PT,DL,TS,RV,FS,LV,OT
4	>110 m	FV,SV,TU,WA,WB,OT,CF,CH,CO,DL,FR,LC,OT,PT,RV,TU,WA,WB,FV,SV,SA,CF,CO,FV,OT,FR,LC,CH,PT,DL,TS,RV,FS,LV,OT
5	0 to 180 m	TA,LN,LP,BC,OB
6	181 to 240 m	TA,LN,LP,BC,OB
7	241 to 280 m	TA,LN,LP,BC,OB
8	> 280 m	TA,LN,LP,BC,OB
9	0 to 180 m	SC,CS,CR,LA,CC,C2,C3,RE
10	181 to 240 m	SC,CS,CR,LA,CC,C2,C3,RE
11	> 240 m	SC,CS,CR,LA,CC,C2,C3,RE
12	> 0	BA,YA
13	> 0	CX,OR,DR
14	0 to 120 m	PV
15	> 120 m	PV
16	> 0	FB
17	0 to 120 m	NN,NV
18	> 120 m	NN,NV
19	>0	JU,LU,PM

6.3.3 Vessel generator

6.3.3.1 Vessel arrivals

According to the statistics data in Appendix C, there is a peak of vessel arrivals in the seaport, i.e. 5am-7am. As a result, there is a peak in the total number of vessels in the seaport, i.e. 7am-9am. Ideally, the overall distribution of vessel arrivals should reflect such peaks. However, this conclusion is difficult to validate at current stage, and it is not discussed here.

In the simulation system, vessel arrivals at an origin (i.e. a node) are supposed to follow predefined distribution. It is assumed that the arrival at each node satisfy a Poisson distribution (Huang *et al.* 2011). At a node, the arrival intervals are calculated as the interval between any two subsequent vessels. Average of the arrival interval is collected in Table 6.7. From the table,

- Only the node with region names can have vessel arrivals, including anchorage nodes, the berthing nodes and the boarding nodes.
- Vessels are to be created on only 52 nodes. 17 nodes with region names do not have any arrival data. For example, NODE78 is a berthing node with 24 region names. However, in the input data, no vessel travelled from this node, thus no arrival interval is calculated for NODE78.
- Some average arrival intervals are very huge. For example, the average arrival interval of NODE94 is 89.3 hours. When creating schedules for a whole day, there may be no vessel to be created from these nodes. For average arrival interval bigger than 12 hours, there is at most one vessel to be created.

Table 6.7 The average arrival interval.

ID	seconds	minutes	hours	ID	seconds	minutes	hours	ID	seconds	minutes	hours
1	15780	263	4.4	22	14303	238.4	4	71	45023	750.4	12.5
2	13022	217	3.6	23	33500	558.3	9.3	72	28968	482.8	8
3	59965	999.4	16.7	24	5961	99.4	1.7	74	14319	238.7	4
4	8738	145.6	2.4	25	3021	50.4	0.8	75	71023	1183.7	19.7
5	7368	122.8	2	26	137862	2297.7	38.3	76	3451	57.5	1
6	14105	235.1	3.9	27	5637	94	1.6	80	15019	250.3	4.2
7	3938	65.6	1.1	29	24882	414.7	6.9	81	24137	402.3	6.7
9	42710	711.8	11.9	31	62556	1042.6	17.4	84	39033	650.6	10.8
10	39343	655.7	10.9	32	43472	724.5	12.1	94	321445	5357.4	89.3
11	65346	1089.1	18.2	35	11049	184.2	3.1	96	2992	49.9	0.8
12	69941	1165.7	19.4	60	24107	401.8	6.7	102	21305	355.1	5.9
13	58622	977	16.3	63	5528	92.1	1.5	104	98995	1649.9	27.5
14	36778	613	10.2	64	7287	121.5	2	106	198346	3305.8	55.1
15	17271	287.9	4.8	65	26357	439.3	7.3	108	301000	5016.7	83.6
17	680200	11336.7	188.9	66	23806	396.8	6.6	115	4545	75.8	1.3
19	14370	239.5	4	67	163800	2730	45.5	117	2321	38.7	0.6
20	14564	242.7	4	68	196674	3277.9	54.6				
21	3770	62.8	1	70	151046	2517.4	42				

6.3.3.2 Vessel types

At an origin, i.e. a node, vessels are generated with different types. For example, NODE1 is a berthing node. It generates the vessels with types 1,5,12,2,14,13,7,3,4,16,9 and 6. NODE71 corresponds to an anchorage area. All vessels starting at this node are of type 5. Then, the percentage of each vessel type appearing on this node can be calculated.

Table 6.8 lists the percentages of vessel types over the 52 nodes.

Table 6.8 The distribution of vessel types.

ID	Percentages of vessel types
1	1(35.71%), 5(15.48%), 12(25.60%), 2(10.12%), 14(0.60%), 13(4.76%), 7(1.19%), 3(0.60%), 4(0.60%), 16(4.17%), 9(0.60%), 6(0.60%)
2	12(21.95%), 6(2.44%), 1(26.34%), 2(16.10%), 5(20.98%), 13(4.39%), 7(2.44%), 4(0.98%), 16(1.95%), 8(0.98%), 3(1.46%)
3	1(23.81%), 12(26.19%), 13(21.43%), 2(2.38%), 8(2.38%), 5(7.14%), 7(4.76%), 6(4.76%), 9(2.38%), 4(2.38%), 10(2.38%)
4	1(30.92%), 12(15.79%), 16(40.79%), 5(5.26%), 2(3.62%), 13(1.32%), 3(1.64%), 6(0.33%), 9(0.33%)
5	16(53.12%), 4(0.54%), 5(14.36%), 12(9.21%), 1(14.36%), 2(3.52%), 13(2.71%), 7(0.54%), 3(1.08%), 6(0.54%)
6	5(13.16%), 16(12.11%), 1(57.37%), 3(0.53%), 2(3.68%), 14(1.58%), 12(4.74%), 13(2.63%), 4(2.11%), 9(1.58%), 10(0.53%)
7	1(29.61%), 16(64.44%), 12(0.58%), 19(0.58%), 14(2.32%), 5(1.60%), 15(0.29%), 2(0.29%), 13(0.15%), 3(0.15%)
9	6(6.67%), 5(43.33%), 1(20.00%), 12(16.67%), 2(6.67%), 13(5.00%), 3(1.67%)
10	5(73.77%), 7(3.28%), 2(6.56%), 4(4.92%), 6(3.28%), 12(4.92%), 1(1.64%), 3(1.64%)
11	5(74.36%), 6(17.95%), 13(2.56%), 7(5.13%)
12	3(11.11%), 1(5.56%), 5(52.78%), 4(5.56%), 6(2.78%), 12(5.56%), 2(2.78%), 16(11.11%), 14(2.78%)
13	6(20.45%), 5(54.55%), 7(6.82%), 3(6.82%), 4(2.27%), 2(9.09%)
14	5(84.51%), 6(8.45%), 4(2.82%), 7(4.23%)
15	14(42.68%), 15(40.13%), 2(3.82%), 3(3.82%), 5(2.55%), 9(4.46%), 4(1.91%), 13(0.64%)
17	8(66.67%), 5(33.33%)
19	16(18.09%), 7(1.06%), 5(15.96%), 1 (14.89%), 12(13.30%), 14(6.38%), 2(5.32%), 9(1.60%), 8(0.53%), 6(4.79%), 13(4.26%), 3(2.66%), 4(6.91%), 15(1.06%), 10(3.19%)
20	1(25.13%), 12(15.51%), 5(14.97%), 4(4.28%), 13(16.04%), 8(1.07%), 2(12.83%), 9(4.81%), 6(2.14%), 10(1.07%), 11(0.53%), 14(0.53%), 7(0.53%), 16(0.53%)
21	5(93.08%), 13(6.50%), 12(0.28%), 2(0.14%)
22	5(72.83%), 6(13.04%), 7(7.61%), 2(0.54%), 8(4.89%), 4(1.09%)
23	1(33.33%), 5(33.33%), 12(33.33%)
24	6(2.61%), 4(6.97%), 2(45.10%), 12(13.29%), 1(12.85%), 9(4.79%), 5(5.45%), 3(3.70%), 14(3.92%), 19(0.44%), 13(0.87%)
25	9(18.00%), 11(4.40%), 5(18.90%), 4(29.60%), 13(1.10%), 10(8.80%), 3(8.20%), 6(8.70%), 7(0.90%), 2(1.10%), 8(0.10%), 12(0.20%)
26	5(94.12%), 11(5.88%)
27	3(10.00%), 2(10.20%), 5(28.78%), 4(18.57%), 9(18.37%), 1(5.92%), 6(4.49%), 16(1.22%), 10(1.22%), 12(0.82%), 11(0.41%)
29	5(74.29%), 6(10.48%), 7(10.48%), 8(0.95%), 4(0.95%), 12(0.95%), 1(1.90%)
31	5(36.11%), 9(2.78%), 1(41.67%), 6(13.89%), 2(2.78%), 4(2.78%)
32	9(6.78%), 6(25.42%), 5(40.68%), 4(10.17%), 10(5.08%), 8(5.08%), 7(6.78%)
35	5(54.92%), 3(3.69%), 7(2.05%), 12(4.51%), 13(3.69%), 8(1.23%), 9(6.15%), 2(4.92%), 4(4.92%), 6(11.07%), 10(2.05%), 11(0.41%), 16(0.41%)
60	9(3.65%), 2(7.30%), 1(45.26%), 12(31.39%), 15(4.38%), 5(1.46%), 8(2.92%), 13(0.73%), 3(0.73%), 11(2.19%)
63	5(83.37%), 6(6.37%), 13(0.41%), 7(1.03%), 3(3.90%), 2(2.05%), 4(2.67%), 12(0.21%)
64	5(14.44%), 1(40.94%), 12(16.54%), 2(22.57%), 6(1.57%), 3(1.31%), 7(0.26%), 9(0.79%), 4(0.79%), 13(0.79%)
65	1(51.46%), 12(45.63%), 2(1.94%), 9(0.97%)
66	5(88.07%), 6(6.42%), 7(2.75%), 8(1.83%), 13(0.92%)
67	18(66.67%), 5(33.33%)
68	5(9.09%), 1(90.91%)
70	5(29.41%), 2(58.82%), 1(11.76%)
71	5(100.00%)
72	5(56.82%), 7(12.50%), 6(22.73%), 8(5.68%), 10(1.14%), 4(1.14%)
74	12(24.70%), 19(63.75%), 2(3.59%), 1(3.98%), 16(3.59%), 5(0.40%)
75	13(34.62%), 2(11.54%), 12(11.54%), 9(7.69%), 1(11.54%), 5(15.38%), 16(7.69%)
76	2(48.55%), 1(17.30%), 13(5.58%), 3(1.56%), 12(19.87%), 16(1.67%), 9(0.67%), 5(2.23%), 4(1.79%), 10(0.33%), 6(0.11%), 11(0.11%), 14(0.22%)
80	5(38.64%), 16(45.45%), 4(1.70%), 6(7.95%), 7(2.27%), 12(2.84%), 1(0.57%), 13(0.57%)
81	12(37.61%), 1(46.79%), 2(14.68%) ,13(0.92%)
84	13(6.25%), 5(18.75%), 6(9.38%), 18(3.13%), 3(6.25%), 4(9.38%), 14(1.56%), 10(4.69%), 9(26.56%), 2(9.38%), 7(1.56%), 8(1.56%), 17(1.56%)
94	13(14.29%), 5(28.57%), 2(42.86%), 3(14.29%)
96	12(42.20%), 1(39.82%), 2(3.00%), 13(14.71%), 16(0.26%)
102	2(50.74%), 1(19.85%), 12(15.44%), 19(11.76%), 4(0.74%), 16(0.74%), 13(0.74%)
104	3(11.54%), 18(7.69%), 9(38.46%), 4(7.69%), 5(15.38%), 2(19.23%)
106	1(76.92%), 2(7.69%), 7(7.69%), 18(7.69%)
108	5(83.33%), 4(16.67%)
115	1(28.08%), 12(29.28%), 2(14.99%), 5(7.85%), 9(0.76%), 6(2.23%), 3(0.49%), 7(0.90%), 13(5.27%), 8(1.03%), 19(4.37%), 16(2.53%), 14(0.68%), 4(1.11%), 15(0.05%), 10(0.30%), 11(0.08%)
117	9(10.12%), 5(5.62%), 11(1.98%), 4(8.00%), 12(24.14%), 1(20.44%), 2(6.08%), 3(5.56%), 7(1.32%), 13(9.39%), 10(2.71%), 6(3.57%), 8(0.66%), 18(0.26%), 19(0.07%), 17(0.07%)

6.3.3.3 Generation of routes

A vessel may need to perform several tasks in the seaport. For example, it may need to unload the cargos, do some repairing work, load other cargos and leave the seaport. Thus, it travels from one node to another in a sequence. All vessels setting out at a same node may target different nodes to perform different tasks. Thus, each node can also accumulate the times for different routes that had been used in the input schedule, and calculate the probability for each route. As mentioned in Section 6.3.2.3, the simulation system inputs 901 pre-defined routes. Statistical results show that the routes with IDs smaller than 638 are used in real time. In Table 6.9, On the NODE15, most of the vessels took the ROUTE190, which is from NODE15 to NODE7. On the NODE17, all vessels adopted ROUTE194, which leads them to NODE22.

Table 6.9 Percentage of routes at different nodes.

ID	Percentage of routes
1	5(5.95%), 15(8.33%), 10(16.07%), 9(5.95%), 12(3.57%), 1(17.86%), 2(3.57%), 11(5.36%), 14(4.76%), 6(4.17%), 3(5.95%), 7(3.57%), 8(1.19%), 4(12.50%), 13(0.60%), 16(0.60%)
2	21(10.73%), 32(8.78%), 18(14.63%), 38(0.98%), 37(4.88%), 31(20.98%), 36(2.44%), 35(1.95%), 29(0.98%), 39(6.83%), 20(3.41%), 40(0.98%), 34(7.32%), 30(6.83%), 22(0.98%), 27(1.46%), 33(1.46%), 19(0.49%), 23(0.98%), 24(0.98%), 25(0.98%), 28(0.49%), 26(0.49%)
3	42(21.43%), 49(16.67%), 43(14.29%), 53(14.29%), 45(4.76%), 52(9.52%), 51(2.38%), 54(2.38%), 46(2.38%), 47(2.38%), 48(2.38%), 44(4.76%), 50(2.38%)
4	60(15.46%), 61(35.53%), 56(7.24%), 69(12.50%), 68(9.21%), 64(2.30%), 65(1.32%), 62(2.96%), 57(2.96%), 66(0.99%), 59(6.58%), 63(1.64%), 58(0.99%), 67(0.33%)
5	76(57.45%), 78(1.90%), 74(5.69%), 80(1.90%), 77(4.07%), 83(8.94%), 84(0.27%), 81(4.07%), 87(3.25%), 86(1.08%), 71(3.25%), 72(3.79%), 85(1.36%), 75(1.63%), 88(0.27%), 82(0.54%), 79(0.27%), 73(0.27%)
6	92(7.37%), 94(63.68%), 104(2.11%), 97(2.63%), 98(8.42%), 103(1.05%), 90(2.11%), 96(1.05%), 101(2.11%), 91(1.05%), 100(1.58%), 93(3.16%), 95(2.11%), 99(0.53%), 102(1.05%)
7	110(16.69%), 108(24.24%), 109(37.30%), 106(1.45%), 111(3.19%), 122(4.50%), 120(1.60%), 117(2.03%), 116(0.15%), 113(6.24%), 112(0.15%), 107(0.15%), 121(1.60%), 114(0.15%), 118(0.29%), 115(0.15%), 119(0.15%)
9	125(1.67%), 132(15.00%), 124(10.00%), 133(21.67%), 127(16.67%), 128(25.00%), 131(1.67%), 129(3.33%), 130(3.33%), 126(1.67%)
10	142(37.70%), 141(22.95%), 140(18.03%), 135(4.92%), 143(1.64%), 137(4.92%), 136(1.64%), 139(1.64%), 138(4.92%), 144(1.64%)
11	150(48.72%), 149(41.03%), 148(2.56%), 146(2.56%), 151(2.56%), 147(2.56%)
12	161(36.11%), 159(2.78%), 155(11.11%), 160(2.78%), 153(5.56%), 163(2.78%), 162(13.89%), 158(5.56%), 156(13.89%), 157(2.78%), 154(2.78%)
13	172(47.73%), 168(4.55%), 171(15.91%), 169(11.36%), 173(2.27%), 170(2.27%), 166(6.82%), 165(4.55%), 167(4.55%)
14	188(33.80%), 182(5.63%), 185(11.27%), 178(5.63%), 183(25.35%), 180(5.63%), 184(2.82%), 175(2.82%), 181(1.41%), 176(1.41%), 177(1.41%), 179(1.41%), 186(1.41%)
15	190(80.25%), 191(14.65%), 189(2.55%), 192(2.55%)
17	194(100.00%)
25	300(65.10%), 299(2.60%), 305(6.00%), 303(10.00%), 292(1.80%), 293(1.60%), 291(4.20%), 302(3.10%), 301(2.00%), 296(0.90%), 298(1.10%), 297(0.70%), 304(0.10%), 295(0.30%), 306(0.10%), 294(0.30%), 290(0.10%)
26	307(11.76%), 309(70.59%), 308(11.76%), 310(5.88%)
27	335(60.82%), 332(5.92%), 324(1.02%), 315(5.71%), 325(2.04%), 326(2.86%), 316(9.18%), 314(3.27%), 311(0.61%), 318(1.02%), 312(0.82%), 329(1.02%), 334(0.20%), 331(0.82%), 327(0.61%), 319(0.20%), 328(1.43%), 317(0.82%), 321(0.20%), 313(0.20%), 322(0.61%), 320(0.20%), 330(0.41%)
29	337(5.71%), 341(0.95%), 344(32.38%), 343(44.76%), 342(1.90%), 339(6.67%), 338(3.81%), 340(0.95%), 336(2.86%)
31	347(25.00%), 348(33.33%), 349(19.44%), 346(22.22%)
32	359(61.02%), 354(6.78%), 356(13.56%), 358(6.78%), 351(3.39%), 357(1.69%), 352(3.39%), 353(1.69%), 355(1.69%)

Table 6.9 Percentage of routes at different nodes (continued).

ID	Percentage of routes
35	371(8.61%), 369(5.74%), 363(3.28%), 364(4.92%), 361(7.79%), 374(18.03%), 360(3.69%), 372(23.36%), 370(2.46%), 376(7.38%), 367(5.74%), 365(2.46%), 368(2.87%), 375(0.82%), 377(0.82%), 373(1.23%), 362(0.41%), 366(0.41%)
60	385(8.03%), 379(23.36%), 384(11.68%), 387(33.58%), 383(2.92%), 382(4.38%),380(11.68%), 388(2.19%), 381(1.46%), 386(0.73%)
63	407(24.02%), 394(10.27%), 409(25.67%), 397(5.54%), 393(1.44%), 399(5.95%), 398(6.78%), 403(5.95%), 391(2.67%), 400(5.54%), 390(1.23%), 395(1.03%), 406(0.82%), 404(0.62%), 402(0.62%), 396(1.03%), 405(0.21%), 401(0.41%), 392(0.21%)
64	415(1.84%), 416(5.25%), 410(12.60%), 414(18.37%), 419(5.25%), 426(5.77%), 423(1.05%), 413(12.07%), 420(3.67%), 412(9.19%), 417(0.79%), 411(11.55%), 428(6.82%), 418(0.26%), 422(2.62%), 421(2.36%), 424(0.52%)
65	433(3.88%), 435(84.47%), 432(1.94%), 431(4.85%), 430(2.91%), 429(1.94%)
66	438(5.50%), 437(17.43%), 448(5.50%), 450(16.51%), 443(25.69%), 442(2.75%), 446(5.50%), 444(5.50%), 445(7.34%), 440(2.75%), 439(1.83%), 447(0.92%), 441(1.83%), 436(0.92%)
67	452(33.33%), 451(33.33%), 453(33.33%)
68	456(9.09%), 457(72.73%), 455(9.09%), 454(9.09%)
70	462(17.65%), 458(5.88%), 459(5.88%), 464(52.94%), 461(11.76%), 460(5.88%)
71	466(80.33%), 465(16.39%), 468(1.64%), 467(1.64%)
72	476(5.68%), 475(5.68%), 482(51.14%), 480(1.14%), 469(3.41%), 474(2.27%), 472(1.14%), 478(14.77%), 473(1.14%), 470(2.27%), 471(6.82%), 479(2.27%), 481(1.14%), 477(1.14%)
74	491(89.24%), 485(3.59%), 484(3.19%), 489(1.99%), 486(0.80%), 487(0.80%), 488(0.40%)
75	492(15.38%), 497(11.54%), 501(3.85%), 494(15.38%), 495(11.54%), 496(7.69%), 499(11.54%), 493(15.38%), 500(3.85%), 498(3.85%)
76	521(81.81%), 506(3.35%), 504(1.56%), 503(1.79%), 515(1.56%), 513(1.34%), 509(2.01%), 505(0.45%), 514(1.90%), 507(1.34%), 512(0.11%), 508(0.67%), 517(0.22%), 510(0.56%), 516(0.45%), 519(0.33%), 518(0.45%), 511(0.11%)
80	537(33.52%), 527(34.09%), 535(1.70%), 534(7.95%), 530(0.57%), 526(9.09%), 532(3.98%), 525(1.14%), 531(2.84%), 523(1.14%), 529(0.57%), 533(0.57%), 524(2.27%), 528(0.57%)
81	539(20.18%), 538(13.76%), 549(23.85%), 543(4.59%), 541(12.84%), 545(1.83%), 540(10.09%), 546(0.92%), 544(3.67%), 542(0.92%), 547(7.34%)
84	572(12.50%), 574(23.44%), 573(46.88%), 575(10.94%), 569(1.56%), 576(3.13%), 570(1.56%)
94	581(14.29%), 579(57.14%), 578(28.57%)
96	586(99.30%), 582(0.09%), 584(0.62%)
102	592(95.59%), 590(4.41%)
104	596(84.62%), 594(15.38%)
106	599(92.31%), 598(7.69%)
108	600(66.67%), 602(33.33%)
115	603(13.25%), 620(29.11%), 608(3.23%), 618(17.27%), 606(3.01%), 615(9.75%), 616(0.60%), 613(6.22%), 619(4.83%), 612(5.05%), 604(0.87%), 609(2.80%), 610(1.06%), 605(0.43%), 607(1.47%), 617(0.60%), 611(0.08%), 614(0.35%)
117	628(14.75%), 626(22.09%), 634(51.12%), 630(1.98%), 636(3.70%), 638(0.79%), 633(0.93%), 632(3.97%), 631(0.33%), 629(0.07%), 637(0.20%), 627(0.07%)

6.3.3.4 Generation of schedules

All schedules used in our system are generated according to the statistics data calculated from the input schedules. Different schedules are generated at different nodes independently. On each node, a sequence of schedules is generated according to the distribution of its arrival interval. For each schedule, the vessel type is defined based on the distribution of vessel types on this node. The route for this vessel is obtained using the distribution of routes on the node.

By default, a schedule chart contains the schedules for a whole day. Then, the simulation system will create a set of schedule charts. For sensitivity analysis, 100 schedule charts will be generated for each analysis.

6.3.3.5 Perturbation of vessel schedule

Once a schedule chart is generated, preplanning is applied before it goes to simulation. According to the discussion in the previous chapters, in real life situation, vessels may not exactly follow the schedule. However, a vessel may be behind or ahead of the schedule. In the simulation, there is a perturbation in the vessels' arrival time. Again, such perturbation should follow certain distribution. However, since we do not have any input data on such perturbation, the system adopts a uniform distribution with an interval of 10 minutes.

6.3.4 Sensitivity analysis

Sensitivity analysis is performed for investigation of an influence factor of conflicts: traffic volume, which aims to prove that

- Effectiveness and robustness of the conflict resolution measures under high traffic volume;
- A reasonable change of conflicts when traffic volume is changed, in regards with the number of conflicts and total delays resulted from conflict resolution measures.

6.3.4.1 Traffic volume

The traffic volume is the total number of vessels in the seaport. In the simulation system, the traffic volume is depending on the arrival interval on each node. As the arrival interval on each node increases/decreases, the traffic volume will decrease/increase accordingly. According to the changes of the traffic volume, this study is conduced to see how the preplanning and simulation results are affected. In this study, the traffic volume is controlled by a rate value r. Suppose the arrival interval on the i-th node is a_i. After change the traffic volume by a rate r, the arrival interval a_i becomes $a_i/(1+r)$. If the traffic volume increases by r, all arrival intervals decrease, thus the total number of vessels increases. For example, $r = 10\%$ means that the traffic volume is increased by 10%. The arrival interval a_i decreases to be $a_i/1.1$.

For each rate r, 100 schedule charts are created according to the new arrival interval. Preplanning and simulation are applied on each schedule chart. Each schedule chart provides a result data set (Table 6.10):

- the number of vessels,
- the total delay in preplanning,
- the number of conflicts before applying preplanning,
- the number of conflicts after preplanning, and
- the total delay in the simulation.

Thus, for each rate *r*, 100 result data sets are collected, and a result data set is obtained as the average of all 100 result data sets.

Table 6.11 lists three average data sets for three rates.

Table 6.10 Three result data sets.

Vessels	**Delayed vessels**	**Total preplanning delay (seconds)**	**Conflicts before preplanning**	**Conflicts after preplanning**	**Total simulation delays (seconds)**
69	16	12905	22	6	2201
71	14	27539	15	5	975
76	18	45033	15	2	705

Table 6.11 Average data sets for three rates.

Rate (%)	**Vessels**	**Delayed vessels**	**Total preplanning delay (seconds)**	**Conflicts before preplanning**	**Conflicts after preplanning**	**Total simulation delays (seconds)**
-20	72	15	21280	19	6	1030
-19	72	16	24885	20	7	1076
-18	74	17	26917	20	7	1051

6.3.4.2 Traffic volume over the seaport

This section shows the analysis over the seaport. The statistics for different traffic volume rates are collected in Table 6.12. Figure 6.14 plots the statistics according to the rate. Figure 6.15 plots the changes in the number of vessels and the number of conflicts according to the rate. As expected, the following conclusions are derived:

- As the rate *r* changes from -20% to 20%, the number of vessels also changes from -20% to 20% (Figure 6.15).
- The number of conflicts changes from about -37% to 40%, which is about twice as that of the rate *r* (Figure 6.15).
- The total delay in preplanning *T* and the number of delayed vessels *N* are increase as the rate *r* increases. The average preplanning delay is the defined as the total preplanning delay divided by the number of delayed vessels: *T*/*N*. It also increases accordingly. Thus, as the traffic volume increases, the preplanning will need to delay more vessels, and, on average, the preplanning will place more delay on each delayed vessel.
- The total simulation delay will not change that much according to the rate *r*. This is due to that the preplanning results are feasible. The preplanning already resolves most of the conflicts, and the simulation can predict all remain conflicts. These conflicts are resolved in the simulation.

Table 6.12 Statistics over the seaport.

Rate (%)	Vessels	Delayed vessels	Total preplanning delay (seconds)	Conflicts before preplanning	Conflicts after preplanning	Total simulation delays (seconds)
-20	310	146	930143	110	40	1800
-19	313	146	889104	109	40	1427
-18	316	149	915136	113	38	1487
-17	320	151	976429	117	39	1182
-16	324	155	1007488	122	40	1674
-15	328	159	1095902	126	43	1448
-14	331	161	1139306	127	45	1752
-13	335	162	1127539	124	45	1576
-12	339	166	1178083	131	44	1866
-11	342	168	1187048	133	46	1326
-10	346	170	1245650	140	48	1307
-9	349	172	1256305	137	46	1318
-8	354	177	1360763	147	46	1423
-7	358	181	1380213	145	48	1463
-6	361	182	1421786	154	52	1328
-5	365	185	1431461	152	50	1519
-4	369	188	1510437	152	52	1448
-3	372	191	1559726	158	51	1489
-2	376	195	1619392	160	53	1738
-1	380	197	1666287	168	53	1587
0	384	200	1689328	172	56	1344
1	387	203	1754114	174	55	1338
2	391	207	1776098	173	58	1562
3	395	210	1867289	182	58	1582
4	398	212	1878906	183	59	1454
5	402	214	1954554	184	60	1569
6	406	218	1993792	193	62	1617
7	410	221	2047213	195	62	1456
8	413	224	2112569	203	63	1288
9	417	228	2197846	201	64	1444
10	421	229	2192013	208	66	1465
11	425	231	2272336	207	64	1434
12	428	235	2303201	215	68	1457
13	432	242	2416605	216	67	1429
14	436	246	2507932	225	69	1446
15	439	242	2422419	222	69	1271
16	442	250	2547247	230	69	1279
17	447	253	2622315	231	72	1199
18	450	255	2666865	233	72	1298
19	454	259	2752747	243	76	1507
20	457	261	2762064	241	76	1495

6.3.4.3 Critical regions

A critical region is a geographic area of water where potential conflicts may often occur. Researchers (Ng, Lim *et al.* 1998) indentify five critical regions (see Figure 6.11) within the Port of Singapore, where traffic incidents and accidents are prone to occur,

- Region 1 includes the Eastern Boarding Ground A, a part of the East Keppel Fairway and a part of the Southern Fairway.
- Region 2 includes the Southern Boarding Ground, a part of the Jong Fairway and a part of the Southern Fairway.
- Region 3 includes the fairways close to Pulau Senang.
- Region 4 includes the Western Boarding Ground, a part of the Sinki Fairway, and a part of Temasek Fairway.
- Region 5 includes a part of the East Jurong Channel, a part of the Sinki Fairway, and a part of the West Keppel Fairway.

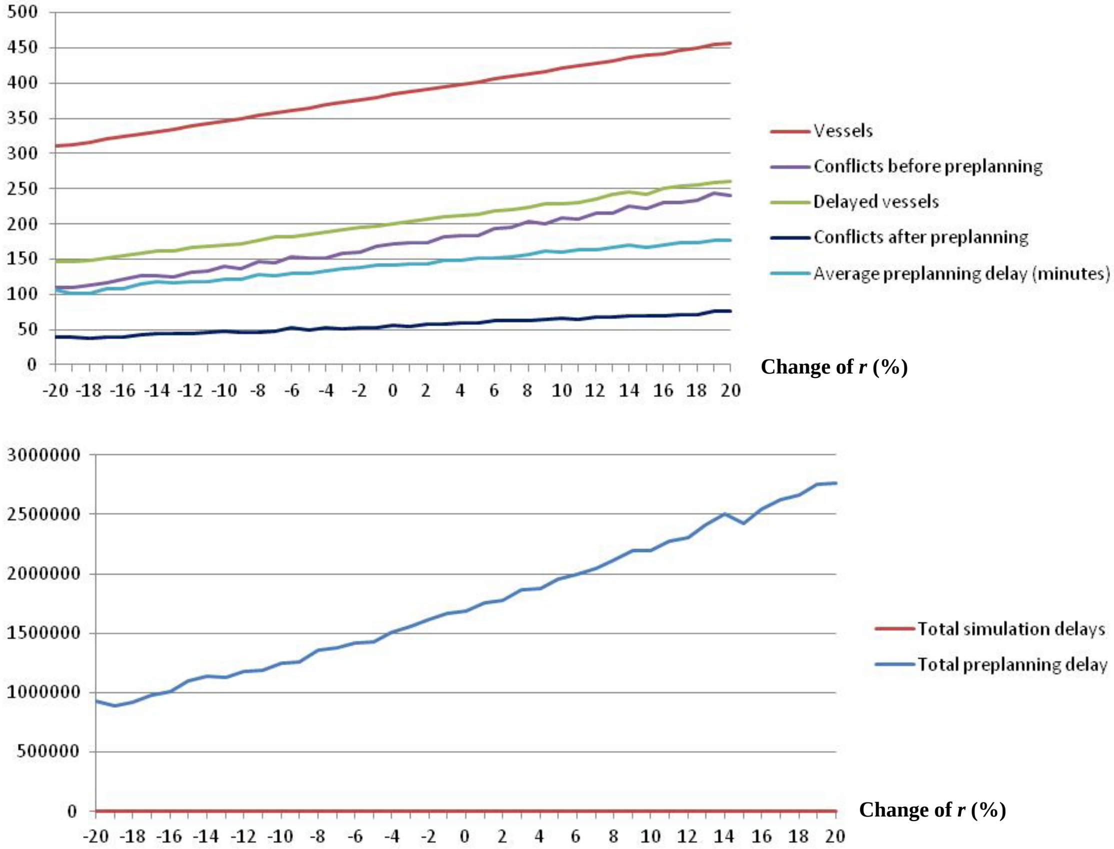


Figure 6.14 The charts for the result in Table 6.12.

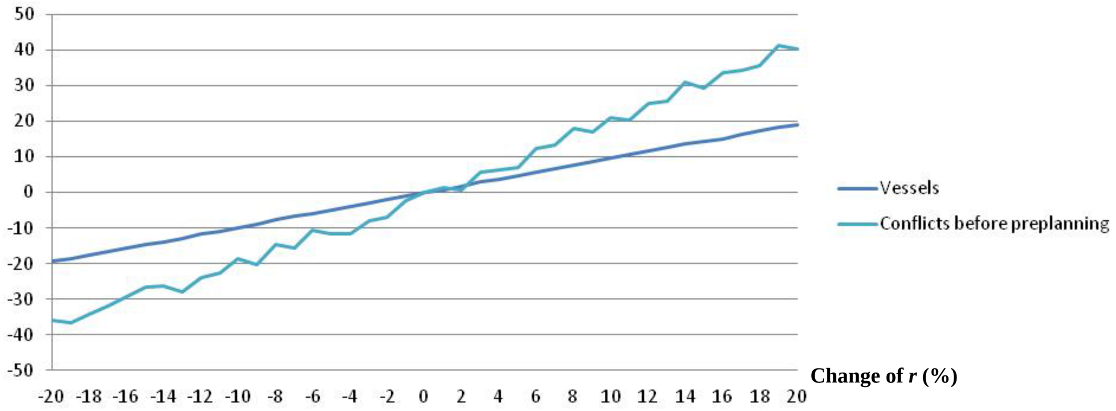


Figure 6.15 The percentage of changes in the vessels and the conflicts in Table 6.12.

In the following, traffic volume analysis will be performed in these critical regions other than the whole traffic network. For each critical region, only the vessels passing/inside the region will be considered.

6.3.4.4 Traffic volume over the critical region 1

This section shows the analysis over the critical region 1. Accordingly, the traffic volume refers to the total number of vessels in this region. The statistics for different traffic volume rates are collected in Table 6.13. Figure 6.16 plots the statistics. We can see that, as traffic volume changes from -20% to 20%, the number of conflicts changes from -35% to 40% correspondingly. The rate of conflict growth is about twice of the growth of the traffic volume. Policy makers can set rules to restrict shipping traffic when it reaches a level that compromises navigation safety.

As expected, the conclusions are similar with that of the whole seaport. As the traffic volume increases, there is a trend in the schedule chart: the number of conflicts increases. The total delay in preplanning also increases accordingly. However, the total delays in the simulation stage change not too much.

Compared with Table 6.12, the statistic data in Table 6.13 are much smaller. The traffic volume in this region is about 23% of the traffic volume of the whole seaport. However, only 17% of the conflicts in the whole seaport occur within this region. The total preplanning delay in this region is only 2.5% of that of the whole seaport. This is because that the preplanning in critical region only considers the vessels that pass the critical region. The preplanning in the whole seaport needs to take care of 83% of the other conflicts appear out of this critical region. Thus, preplanning in the whole seaport will accumulate more delays.

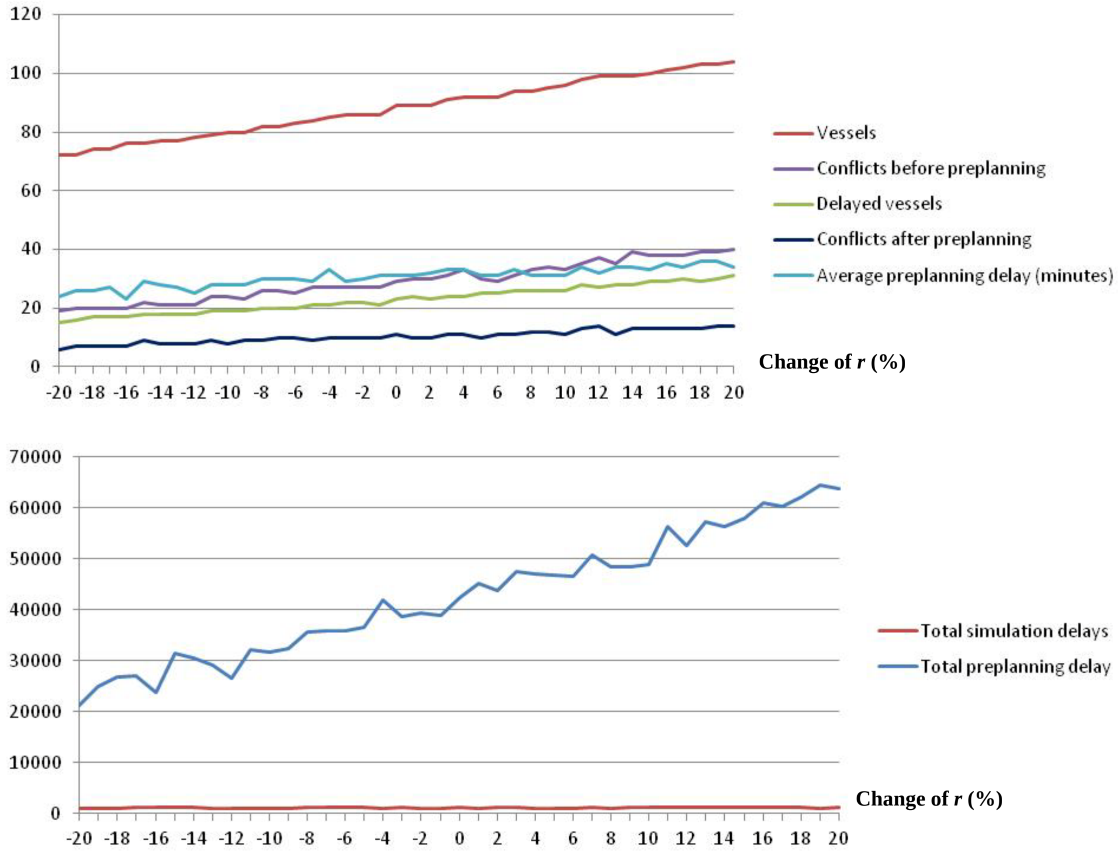


Figure 6.16 The charts of the result in Table 6.13.

Table 6.13 Statistics over the critical region 1.

Rate (%)	Vessels	Delayed vessels	Total preplanning delay (seconds)	Conflicts before preplanning	Conflicts after preplanning	Total simulation delays (seconds)
-20	72	15	21280	19	6	1030
-19	72	16	24885	20	7	1076
-18	74	17	26917	20	7	1051
-17	74	17	27050	20	7	1164
-16	76	17	23872	20	7	1180
-15	76	18	31546	22	9	1158
-14	77	18	30439	21	8	1120
-13	77	18	29192	21	8	1060
-12	78	18	26497	21	8	1029
-11	79	19	32253	24	9	1036
-10	80	19	31643	24	8	1063
-9	80	19	32290	23	9	1097
-8	82	20	35724	26	9	1295
-7	82	20	35767	26	10	1236
-6	83	20	35940	25	10	1149
-5	84	21	36538	27	9	1110
-4	85	21	41847	27	10	1092
-3	86	22	38723	27	10	1116
-2	86	22	39387	27	10	1077
-1	86	21	39011	27	10	1086
0	89	23	42300	29	11	1153
1	89	24	45096	30	10	1078

Table 6.13 Statistics over the critical region 1 (continued).

Rate (%)	Vessels	Delayed vessels	Total preplanning delay (seconds)	Conflicts before preplanning	Conflicts after preplanning	Total simulation delays (seconds)
2	89	23	43731	30	10	1174
3	91	24	47568	31	11	1105
4	92	24	46958	33	11	982
5	92	25	46719	30	10	1097
6	92	25	46624	29	11	1077
7	94	26	50861	31	11	1260
8	94	26	48435	33	12	1061
9	95	26	48368	34	12	1118
10	96	26	48883	33	11	1187
11	98	28	56330	35	13	1175
12	99	27	52629	37	14	1116
13	99	28	57300	35	11	1115
14	99	28	56392	39	13	1152
15	100	29	57895	38	13	1303
16	101	29	60900	38	13	1200
17	102	30	60330	38	13	1183
18	103	29	62212	39	13	1183
19	103	30	64438	39	14	1046
20	104	31	63845	40	14	1120

6.3.4.5 Traffic volume over the critical region 2

The results of the simulation by changing the traffic volume over the critical region 2 are presented in Table 6.14 and Figure 6.17. Similar conclusions as the whole seaport and the critical region 1 are obtained. As the traffic volume increases, the number of vessels, the number of conflicts, and the total delay in the preplanning stage increase accordingly. On the other hand, the delay in the simulation stage hardly remains unchanged.

There are some differences between the critical region 1 and the critical region 2. The traffic volume in the critical region 2 is bigger than the traffic volume in the critical region 1. It is 32% of the traffic volume of the whole seaport. Consequently, the number of conflicts in the critical region 2 is much bigger, and it reaches 36% of the conflicts in the whole seaport. Before the preplanning, as traffic volume changes from -20% to 20%, the number of conflicts changes from -30% to 40%. It is noticed that, the rate of conflict growth is almost the same as the critical region 2 where the number of conflicts changes from -35% to 39%. It would be helpful for policy makers to develop a traffic bottleneck analysis and investigate saturated traffic status in these critical regions. Special rules to restrict shipping traffic are required before traffic volume reaches a level that compromises navigation safety.

Table 6.14 Statistics over the critical region 2.

Rate (%)	Vessels	Delayed vessels	Total preplanning delay (seconds)	Conflicts before preplanning	Conflicts after preplanning	Total simulation delays (seconds)
-20	101	43	150450	41	13	1290
-19	103	43	153435	40	13	1289
-18	104	44	157036	41	13	1199
-17	104	44	164264	42	13	1250
-16	106	46	166511	44	13	1360
-15	109	47	178124	45	13	1291
-14	108	47	175642	46	12	1228
-13	110	49	182626	47	15	1220
-12	111	50	191726	50	15	1331
-11	112	50	198045	48	14	1245
-10	112	50	199095	49	16	1222
-9	115	52	215400	51	16	1226
-8	115	53	215856	52	17	1262
-7	117	54	225925	54	16	1368
-6	119	56	245657	56	15	1298
-5	118	54	239366	54	17	1268
-4	121	58	262282	58	16	1306
-3	122	58	258704	59	17	1298
-2	124	59	275595	61	19	1191
-1	124	59	273426	59	17	1365
0	126	61	287701	59	18	1306
1	126	60	288376	61	19	1166
2	128	62	293705	65	17	1262
3	129	62	295034	63	19	1271
4	130	64	314385	64	20	1311
5	132	66	352869	68	19	1238
6	133	65	331259	68	21	1448
7	136	67	360342	70	21	1246
8	135	67	353630	70	21	1291
9	137	69	352909	72	22	1338
10	139	70	390325	72	21	1257
11	140	70	396113	73	24	1274
12	140	71	383972	75	22	1164
13	142	73	430522	77	24	1293
14	142	73	409598	79	22	1353
15	144	74	431338	79	24	1427
16	145	77	458933	81	23	1278
17	146	76	448655	82	24	1330
18	147	76	468057	82	24	1351
19	149	80	504370	87	25	1386
20	149	78	476291	82	25	1302

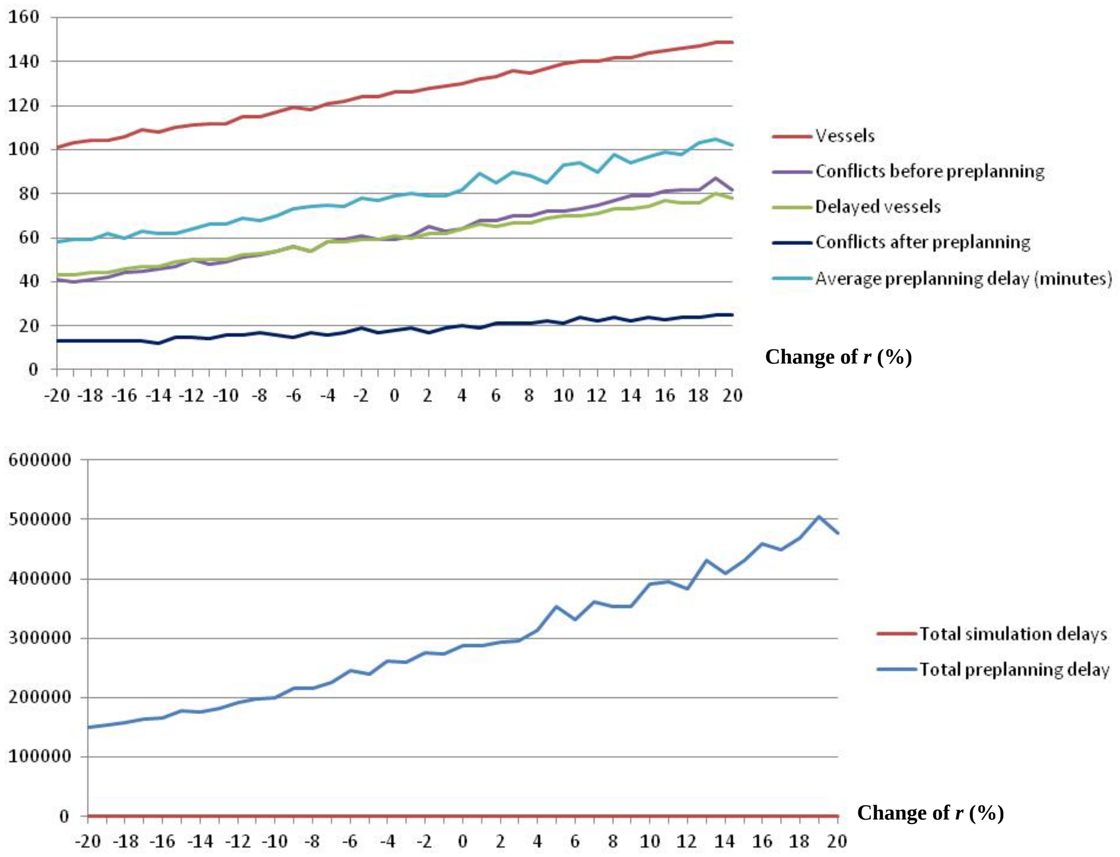


Figure 6.17 The charts of the result in Table 6.14.

6.4 Summary

This chapter presented the overall development of a marine traffic conflict simulation system, which was composed by two parts: system implementation (Section 6.2) and model demonstration (Section 6.3).

Section 6.2 represented the issue of software development including module and graphic interface design, etc. The detail architecture of the simulation system is presented. Five different modules are developed for the simulation system: graphic display, data processing, simulation control, conflict detection, and conflict resolution. The simulator could support most of general seaports characterized by a traffic network consisting of basic elements, e.g. fairways, anchorages and berths etc.

Section 6.3 was a specific demonstration of the Port of Singapore. The data from the Port of Singapore were used as input to the simulation system. Sensitivity analysis over the whole seaport and the critical regions showed that, the number of traffic conflicts is sensitive to the traffic volume. Meanwhile, it was proved that, given any arbitrary schedule chart, the methods for conflict detection and resolution can provide a feasible schedule chart which is free of conflict.

Currently this simulation study has not been extended to other cases of seaports, since quite a lot of the seaport information needed is confidential and sensitive and is protected. It is hard to access to necessary data maintained by external organizations such as port authorities without their cooperation. It is also the reason why validation for the simulation system has not been done. A further work is expected for extending the simulation study to other general seaports.

Chapter 7

Conclusions and future research

This chapter presents a summary of research findings, contributions and some important conclusions. Perspective plans that provide recommendation for further research are presented as well.

7.1 Summary

The main achievement of this research is a systematic conflict resolution strategy for planning for port operators so as to resolve traffic conflicts and congestion ahead of time. Two aspects were considered in conflict resolution. Safe navigation of vessels without conflicts was taken as the first priority. In regards to traffic congestion management, the other issue was focused on the task of delay minimization so as to improve traffic conditions in a whole traffic network. Therefore, the research is useful for marine traffic safety management in seaports and also helps enhancing the efficiency of port operations.

Conflict resolution is a two-stage process. The first stage was referred to as vessel rescheduling for eliminating potential conflicts under original schedules of vessels. An original conflict-free scheduling algorithm was developed in this study for coordinating vessels without conflicts for a given schedule. The second stage focused on a simulation of vessel movements as a result of the stochastic fluctuations in vessel speed and deviations in timetables in real-time. The applicable corrective measures were identified by a decision-making mechanism. These measures conclude general rules or guidance for navigator chooses proper evasive maneuvers.

Correspondingly, the delay minimization was addressed in the preplanning stage and the simulation stage. In the preplanning stage, the delay minimization was solved through searching an optimal schedule chart in the schedule tree. In the simulation stage, Evasive actions would result in delays to vessel operations. Such delays, compared to the delays incurred in the preplanning stage, arise from a dynamic traffic environment which will affect the efficiency of the traffic network. Traffic congestion will appear if delays are accumulated beyond the limitation of tolerable delay in a certain area. Therefore, the decision-making process will check and evaluate that there is sufficient space and time to perform corrective maneuvers while the total delay incurred within the affected area is acceptable.

The conflict prediction algorithm presented another achievement of this research, as an approach to predict a conflict risk multi-link ahead current vessel position. With multi-link-ahead prediction, a possible conflict could be predicted more links ahead of the current vessel position. The required number of links ahead was designed as a variable in the simulation system, and thus could be adjusted for specific requirements. Such conflict prediction as the basis of conflict resolution, guarantees sufficient time for navigator to take actions before the predicted conflict occurs.

This research developed an experimental platform (i.e. Marine Traffic Conflict Simulation System) for evaluating measures for conflict detection and conflict resolution. A study was conducted for

Singapore which is the busiest transshipment hub in the world. On the other hand, simulation of marine traffic conflict is a generic study which is independent of the Port of Singapore, although it was used as an example for demonstration. The logic of conflict detection and resolution is applicable to other traffic systems by changing input data. It is expected to adapt to most of general seaports that are faced with traffic congestion and delays.

7.2 Contributions

The main contributions of this research are summarized as follows.

7.2.1 Systematic strategy for conflict detection and resolution

Marine traffic conflict is a critical issue in navigational safety and traffic congestion management. However, to the best of this researcher' knowledge, no studies are particularly addressed for detecting and resolving vessel conflicts in port waters. A systematic strategy for conflict detection and resolution was presented in this research, referring to a two-step planning process combined with vessel scheduling methods in a deterministic manner and simulation-based guidance for corrective actions in the operational level. The vessel scheduling enables that most of potential conflicts under original arrival schedules could be detected and eliminated. In simulation stage, possible conflicts resulted from the stochastic changes in vessel movements would be predicted using the multi-link look-ahead prediction method. Applicable corrective measures are decided by a decision-making mechanism with general rules or guidance for navigator chooses proper evasive maneuvers.

It is noticed that the logic of conflict detection and resolution is not relying on the traffic pattern of the Port of Singapore, although it was used as an example for demonstration. The methods of conflict detection and resolution are designed for being applicable to general port traffic systems characterized by a traffic network consisting of basic elements, e.g. fairways, anchorages and berths etc. For instance, the multi-link-ahead conflict detection algorithm is developed for predicting a possible conflict between any two approaching vessels. Use of such algorithm is based on a simple assumption: a vessel is given with its safe domain (see Section 4.1), which should be acceptable by most vessels under port operator. Another example is the rescheduling algorithm for conflict-free traffic, which is also independent on the Port of Singapore. The rescheduling algorithm can be used for any seaport characterized by a restricted zone where vessels move along given routes and schedules.

Development of conflict resolution represents a breakthrough in the field of safe and effective marine traffic control (see limitations and gaps in literature in Section 2.3.3). It is a systematic resolution on the practical side for planning of port operators so as to resolve traffic conflicts and congestion ahead of time.

7.2.2 Multi-link look-ahead conflict prediction

Conflict prediction algorithm with multi-link look-ahead is a particular contribution of this research, which has never been considered by other studies. Previous study (see Section 2.1.1) using the criterion of CPAs or ship domain for collision risk determination are developed for determining a collision risk under current two vessels' relative position. Thus, such methods are limited to be applied for prediction a conflict regarding the dynamic change of vessels' relative position. Therefore, this paper proposes a new algorithm for multi-link-ahead prediction for vessel conflicts. With multilink-ahead prediction, a possible conflict can be predicted more links ahead, so as to ensure that navigator has sufficient time to take actions before the predicted conflict occurs.

Simulation results showed that conflicts can be accurately predicted in time. Thus, it can be used in real-time simulation. Theoretically, multi-link-ahead prediction is able to predict all possible conflicts of a vessel at the very beginning when the vessel enters the seaport by setting the required number of links ahead a large value. On the other hand, the larger value of the number of links will resulted in increased computation time. Thus, this value is designed as parameter p_s in the simulation system. By changing the input data, it is flexible to set a proper value of p_s under different requirements of port operator.

7.2.3 Measures of conflict resolution

Conflict free scheduling algorithm

Conflict resolution is a traffic control problem which can be solved with scheduling methods. Development of vessel scheduling is based on a general rule that vessels traveling in a seaport needs to report their schedules to the port control center before their arrival. Thus, the control center is able to coordinate vessel movements through planning of vessel arrivals and departures using scheduling approach.

Conflict free scheduling has been developed in airport ground traffic planning and train station traffic planning (see Section 2.2.2), but never been found in the marine context. In this research a node-based algorithm was designed for preplanning conflict free schedules for a seaport. Simulation results showed that this algorithm can achieve feasible solutions and as well the optimal one in an efficient way. Meanwhile it overcame the drawbacks of infinite loop resulting from other conflict free scheduling algorithms (Fan 1988; Fan and Cao 2000).

Delay minimization

To resolve conflict problem, it is inevitable to bring in time delays for vessels. These delays have impact on the efficiency of port operation, and also are important to evaluate the conflict measures. This thesis presented the issue of delay minimization in the preplanning stage and the simulation stage. In the preplanning stage, optimization was applied to the process of rescheduling, so that the total delay can be minimized subject to restrictions of maximum acceptable delay for each vessel. In the simulation stage, a decision-making was made for checking and evaluate that there is sufficient space and time to perform corrective maneuvers while the total delay incurred within the affected area is acceptable. Delay minimization is theoretically interesting and practically useful in traffic congestion management. This research is an attempt to this issue, on the basis of which further discussions would be done.

Guidance for corrective actions

In the sense of traffic control, proper evasive maneuvers can effectively avoid imminent conflict situations which are urgent events or emergencies. In this thesis, applicable corrective actions were identified by a decision-making mechanism. Beside safety concerns, these measures also ensure that the total delay incurred within the affected area is acceptable. This thesis analyzed a group of conflict scenarios, and provides mathematical models for calculating proper evasion maneuvers in regards with general traffic rules and vessel characteristics. We should notice that, it is hardly possible, in a real encounter, to consider so many variables with expert knowledge to come up with an optimal solution for evasive maneuvers. Thus, the corrective measures addressed in this thesis are considered as guidance for safe navigation, and how the optimal maneuvers are to be conducted by the navigator is not of significance.

7.2.4 Development of marine conflict simulation study

A marine conflict simulation system has been developed using Microsoft Visual Studio under Windows environment. It is an integrated multi-functional simulation platform, consisting of five basic modules: the graphic display module, the data processing module, the simulation control module, the conflict detection module, and the conflict resolution module. Under such modules, the traffic network can be loaded from maps or created by users; vessel arrival can be specified to be certain type of distribution; and the information of vessels can be processed by parameters. It means that the simulator could support most of general seaports characterized by a traffic network consisting of basic elements, e.g. fairways, anchorages and berths etc. This simulation system reflects a work of appealing originality of this research. Appendix D presents core codes for the simulation implementation, *i.e.* C++ classes and functions.

In this research, the role of the simulation system is

- An execution platform for conflict detection and resolution measures: Examples in Section 4.3, Section 5.2.5 and Section 5.3.6, were executed with the simulation system. Simulation provided a graphic illustration for mathematical model, and was of capability in efficient mathematical calculation, as presented in Section 6.3.4.
- An experiential platform for investigating and evaluating of conflict scenarios: In Chapter 6, experiments were conducted using the data collected from the Port of Singapore. Simulation results showed that traffic conflicts can be accurately detected, and effectively and efficiently resolved with the two-stage resolution. Sensitivity analysis for investigation of traffic volume proved the robustness of conflict detection resolution measures and reflected a reasonable relationship between conflict and traffic volume.
- A display platform with dynamic graphics: Simulation technology has good performance in computer animation which presents a direct display of dynamic vessel movements and traffic environment. In this research, conflict scenarios that were displayed visually could aid traffic controller to investigate serious traffic incidents/accidents without involving field trials.

An application model of the simulation system was demonstrated using the seaport of Singapore as an example. Simulation results showed that traffic conflicts could be effectively resolved with combination of the scheduling method and the simulation-based corrective measures. It is noticed that, the simulation is a generic study which can be adapted to other busy seaports that are faced with traffic congestion and delays.

7.3 Research limitations

Although this research has reached its aims, there are some avoidable limitations as follows,

- Verification and validation for the simulation study have not been done due to unavailable real data.
- Conflict scenarios in the simulation system are simple and idealized; complex traffic rules and port operations have not been considered.
- The discussion of delay minimization in the thesis is not sufficiently enough to address this issue. For example, it is significant to identify delay tolerance as acceptable in practice.
- The sensitivity analysis is not robust enough without analyzing other factors which may have influence on conflict results, e.g. vessel speed, fairway width and traffic rules.

7.4 Areas for further research

Marine traffic conflict detection and resolution represents an evolving approach to safe and effective traffic control in port water. This research as one of the pioneer studies in this field has great potential in further improvements and extensions. The recommendations for further works are presented as follows.

7.4.1 Model verification and validation

A simulation model using the example of Port of Singapore was presented in Chapter 6. On the other hand, verification and validation for the model have not been employed due to inaccessible data, which should remain the first priority in future works.

Simulation verification is concerned with correct representation of the computer model. A long term work for system testing, debugging and maintaining, will be conducted as a critical part of the software development procedure. Validation is the determination that the model is accurate representation of the real system. Usual approach to simulation validation refers to the calibration of the model through an iterative comparison between the model and actual system behavior, so as to improve the model. The work in this research is concerned on a general model for traffic conflict detection and resolution in port waters. There is difficulty in accessing data maintained by external organizations such as port authorities without their cooperation, because quite a lot of the seaport information needed is confidential and sensitive and are protected. This would restrict the development of simulation validation for a specific port traffic system.

7.4.2 Sensitivity analysis

In this thesis, the sensitivity analysis was presented for analyzing the relationship between the number of conflicts and the traffic volume. The purpose of the analysis is to prove effectiveness and robustness of the conflict resolution measures under high traffic volume in the simulation system.

During the trials for the sensitivity analysis, the author collected data that may be affected by the traffic volume, including the number of conflicts, the total delay of all the vessels, the average delay of all the vessels, the delay for all the vessels at certain nodes/links, and so on. Reasonable and satisfied results were selected to present in this thesis, e.g. analyses of the relationship between the number of conflicts and the traffic volume. At the same time, the simulator has been executing some other trials, including the relationship between vessel speed and the total delay. Such works is still under progress, and will be furthered in the further work.

In fact, as the traffic volume changes, lots of factors are affected, including the total delay of all the vessels, the average delay of all the vessels, the delay for all the vessels at certain nodes/links, and so on. It will be more theoretically stimulating if a further work is able to come up with a quantitative model to roughly describe the relationship between the delay and traffic volume.

7.4.3 Considerations in conflict resolution

Current measures for conflict resolution are developed based on some assumptions. Possible improvement for conflict resolution in future may focus on the following issues,

- Complicated factors in real traffic environment: In Section 4.1.3, vessels were assumed to travel along a predefined trajectory which is consisting of straight lines and circular arcs. Further work will be concerned with complicated factors, such as human error, speed of wind, and the effect of water, which would affect vessel movement as well as conflict situation.
- Traffic regulations: For example, in narrow channels that are common in port waters, a vessel has to keep as near to the outer limit of the channel which lies on her starboard side; and no vessel is allowed to anchor in a narrow channel (IMCO 1972). In such case, roles of give-way vessel and stand-on vessel in a conflict situation should be identified according to corresponding rules.
- Multi-vessel encounter situation: a conflict focused in this research is an impact between two vessels. In reality, more than two vessels may be involved in a conflict situation (*i.e.* multi-vessel conflict), which often occurs in merging, diverging and crossing points of fairways. The resolution for multi-vessel conflict needs to solve several pairs of two-vessel conflicts one by one. The vessel with higher priority will be solved earlier. The priorities for vessels can be set according to vessel speed, vessel dimensions, vessel characteristics and other factors.
- Delay minimization: this thesis discussed the delay minimization in preplanning stage and simulation stage. The incurred total delays will be analyzed in further for evaluating effectiveness of the conflict resolution.

7.4.4 Extensions to online simulation

The current simulation system is an offline simulation for the planning of traffic control center before vessels' arrival to the port. Application of offline simulation is imitated for information communication and data transmission among different platforms. With development of VTS, it is desirable to extend the simulation platform with online traffic surveillance, vessel tracking and positioning, which may facilitate conflict detection and resolution measures in the real-time manner.

A plan/framework for the online simulation is designed with High Level Architecture (HLA) (see Figure 7.1). Under HLA, each federate unit represents a corresponding functional module in the simulation system. Communication among federate units is managed by a middleware, called Run-Time Infrastructure (RTI). RTI has two kinds of components: Central RTI Component (CRC) and Local RTI Component (LRC). CRC is kept in a specified federate unit, and LRCs are embedded in the interfaces of other federate units. The RTI provides a set of software services that support federate units to coordinate their operations and data exchange.

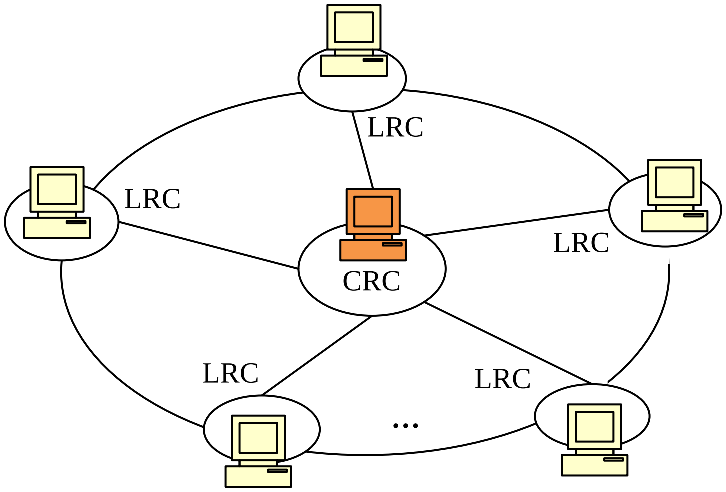


Figure 7.1 Star-form topological structure of RTI.

With the interface accessing CRC, each subsystem can obtain information required from other subsystems, such as environment system, radar system, 3D graphics system and conflict resolution system. Environment system collects data about hydrology condition and weather condition; radar system acquires real-time vessel information (e.g. speed, course); 3D graphics system displays simulation details with dynamic 3D graphics; and conflict resolution system is used to provide measures/actions to navigators.

Reference

Aarsaether, K.G., & Moan, T. (2007). Combined Maneuvering Analysis, AIS and Full Mission Simulation. *International Journal of Marine Navigation and Safety of Sea Transportation*, 1(1), 31-36.

Acar, U., Ziarati, R., & Ziarati, M. (2011). An Investigation into Colregs and Their Applications at Sea. *Bridge 2011 Conference*, Rauma, Finland.

Anderson, D., & Lin, X.G. (1996). A Collision Risk Model for a Crossing Track Separation Methodology. *Journal of Navigation*, 49(3), 337-349.

Browning, A.W. (1991). A Mathematical-Model to Simulate Small Boat Behavior. *Simulation*, 56, 329-336.

Bruzzone, A.G., Mosca, R., Revetria, R., & Rapallo, S. (2000). Risk Analysis in Harbor Environments Using Simulation. *Safety Science*, 35(1-3), 75-86.

Cairns, W.R. (2005). AIS and Long Range Identification & Tracking. *Journal of Navigation*, 58(2), 181-189.

Chang, K.Y., Jan, G.E., & Parberry, I. (2003). A Method for Searching Optimal Routes with Collision Avoidance on Raster Charts. *Journal of Navigation*, 56(3), 371-384.

Chauvin, C., & Lardjane, S. (2008). Decision Making and Strategies in an Interaction Situation: Collision Avoidance at Sea. *Transportation Research Part F: Traffic Psychology and Behaviour*, 11(4), 259-269.

Chin, H.C., & Debnath, A.K. (2009). Modeling Perceived Collision Risk in Port Water Navigation. *Safety Science*, 47(10), 1410-1416.

Clements, D. (1998). High Speed Craft and Colregs - a Proposal. *Journal of Navigation*, 51(3), 430-436.

Cockcroft, A.N. (2003). The Use of VHF in Collision Avoidance at Sea. *Journal of Navigation*, 56(2), 338-340.

Cockcroft, A.N., & Lameijer, J.N.F. (2003). *A Guide to the Collision Avoidance Rules: International Regulations for Preventing Collisions at Sea*, Butterworth-Heinemann.

Coldwell, T.G. (1983). Marine Traffic Behaviour in Restricted Waters. *Journal of Navigation*, 36(3), 430-444.

David, R. (1983). Cargo Ships Routing and Scheduling: Survey of Models and Problems. *European Journal of Operational Research*, 12(2), 119-126.

David, R. (1993). Ship Scheduling: The Last Decade. *European Journal of Operational Research*, 71(3), 325-333.

Davis, P.V., Dove, M.J., & Stockel, C. (1980). A Computer Simulation of Marinetraffic Using Domains and Arenas. *Journal of Navigation*, 33(2), 215-222.

Debnath, A.K., & Chin, H.C. (2010). Navigational Traffic Conflict Technique: A Proactive Approach to Quantitative Measurement of Collision Risks in Port Waters. *Journal of Navigation*, 63(1), 137-152.

Debnath, A.K., Chin, H.C., & Haque, M.M. (2011). Modelling Port Water Collision Risk Using Traffic Conflicts. *Journal of Navigation*, 64(4), 645-655.

Degre, T. (1995). The Management of Marine Traffic, a Survey of Current and Possible Future Measures. *Journal of Navigation*, 48(1), 53-69.

Doniec, A., Mandiau, R., Piechowiak, S., & Espié, S. (2008). A Behavioral Multi-Agent Model for Road Traffic Simulation. *Engineering Applications of Artificial Intelligence*, 21(8), 1443-1454.

Eyferth, K., Niessen, C., & Spaeth, O. (2003). A Model of Air Traffic Controllers' Conflict Detection and Conflict Resolution. *Aerospace science and technology*, 7(6), 409-416.

Fan, H. (1988). Simulation of Aircraft Movements in the Terminal Airspace. *Transportation Planning and Technology*, 13(1), 31-42.

Fan, H.S.L., & Cao, J.M. (2000). Sea Space Capacity and Operation Strategy Analysis System. *Transportation Planning and Technology*, 24(1), 49-63.

Filipowicz, W. (2004). Vessel Traffic Control Problems. *Journal of Navigation*, 57(1), 15-24.

Friis-Hansen, P., & Simonsen, B.C. (2002). Gracat: Software for Grounding and Collision Risk Analysis. *Marine Structures*, 15(4-5), 383-401.

Fujii, Y., & Tanaka, K. (1971). Traffic Capacity. *Journal of Navigation*, 24(4), 543-552.

Furuhashi, T., Nakaoka, K., & Uchikawa, Y. (1996). A Study on Fuzzy Classifier System for Finding Control Knowledge of Multi-Input Systems. *Genetic Algorithms and Soft Computing* (pp. 489-502).

Goodwin, E.M. (1975). A Statictical Study of Ship Domains. *Journal of Navigation*, 28(3), 328-344.

Grossmann, H., Otto, A., Stiller, S., & Wedemeier, J. (2007). Growth Potential for Maritime Trade and Ports in Europe. *Intereconomics*, 42(4), 226-232.

Guo, X., Huang, S.Y., Hsu, W.J., & Low, M.Y.H. (2009). A Simulation Based Hybrid Algorithm for Yard Crane Dispatching in Container Terminals. *Proceedings of the 2009 Winter Simulation Conference (WSC)*, 2320-2331.

Harre, I. (2000). AIS Adding New Quality to VTS Systems. *Journal of Navigation*, 53(3), 527-539.

Hasegawa, K. (2004). Some Recent Developments of Next Generation's Marine Traffic Systems. *IFAC Conference on Control Applications in Marine Systems*, 13-18.

Hasegawa, K., Tashiro, G., Kiritani, S., & Tachikawa, K. (2001). Intelligent Marine Traffic Simulator for Congested Waterways. *IEEE International Conference on Methods and Models in Automation and Robotics*, 631-636.

Hashimoto, A., & Okushima, T. (1990). Evaluating Marine Traffic Safety at Channels. *Accident Analysis and Prevention*, 22(5), 421-442.

Hassan, S.A. (1993). Port Activity Simulation: An Overview. *ACM SIGSIM Simulation Digest*, 23(2), 17-36.

Hatzack, W., & Nebel, B. (2001). The Operational Traffic Control Problem: Computational Complexity and Solutions. *Cesta, & Borrajo (Cesta & Borrajo, 2001)*, 49-60.

Hayama, I., Akio, S.M., Saburo, T., Masaaki, I., & Hisashi, M. (1987). Basic Research on an Expert System for Navigation at Sea. *The Journal of Japan Institute of Navigation*, 77, 133-139.

Hu, S. (2001). Analysis of Anti-Collision Stages During Ships' Encounter (in Chinese). *Navigation of China*, 49(2).

Hu, S., Fang, Q., Xia, H., & Xi, Y. (2007). Formal Safety Assessment Based on Relative Risks Model in Ship Navigation. *Reliability Engineering & System Safety*, 92(3), 369-377.

Huang, S.Y., Hsu, W.J., & He, Y. (2011). Assessing Capacity and Improving Utilization of Anchorages. *Transportation Research Part E: Logistics and Transportation Review*, 47(2), 216-227.

Hughes, T. (1998). Vessel Traffic Services (VTS): Are We Ready for the New Millennium? *Journal of Navigation*, 51(3), 404-420.

IMCO (1972). The International Regulatios for Preventing Collisions at Sea.

Inaishi, M., & Matsumura, H. (1992). Basic Research on a Collision Avoidance System Using Neural Networks. *7th International Congress of IAIN*, Cairo, 221-229.

Ince, A.N., & Topuz, E. (2004). Modelling and Simulation for Safe and Efficient Navigation in Narrow Waterways. *Journal of Navigation*, 57(1), 53-71.

Isbister, E.J., & O'Sullivan, J.P. (1976). Marine Traffic Conflict Assessment System. U.S. Patent, Sperry Rand Corporation.

Jin, Y., Koyama, T., & Zhang, Z. (1990). The Marine Traffic Control System as a Distributed Problem Solving Network. *IEEE International Conference on Systems, Man and Cybernetics*, 876--881.

Judson, B. (1992). Collision Risk Circumstances and Traffic Routeing in the Approaches to the Strait of Defuca,Juan. *Journal of Navigation*, 45(3), 400-413.

Kader, A., Saman, A., Sulaiman, O., Malik, A., Maimun, A., & Saharuddin, A. (2011). Predictive and Reliability Based Collision Risk Aversion Model for Inland Waterways: The Case Frequency Estimation for Malaysian Langat River. *Jurnal Mekanikal*, 32, 38-60.

Kao, S.L., Lee, K.T., Chang, K.Y., & Ko, M.D. (2007). A Fuzzy Logic Method for Collision Avoidance in Vessel Traffic Service. *Journal of Navigation*, 60(1), 17-31.

Kemp, J. (2002). Collision Regulations-Discussion. *Journal of Navigation*, 55(1), 145-155.

Kingsley, F.A. (2006). AIS and Long Range Identification and Tracking. *Journal of Navigation*, 59(1), 167-168.

Kite-Powell, H.L. (2001). Marine Policy: Shipping and Ports. *Encyclopedia of ocean sciences*.

Klein, G.A. (1993). A Recognition-Primed Decision (RPD) Model of Rapid Decision Making. *Decision making in action: Models and methods*, 5(4), 138-147.

Kose, E., Basar, E., Demirci, E., Guneroglu, A., & Erkebay, S. (2003). Simulation of Marine Traffic in Istanbul Strait. *Simulation Modelling Practice and Theory*, 11(7-8), 597-608.

Lam, J.S.L. (2011). Patterns of Maritime Supply Chains: Slot Capacity Analysis. *Journal of Transport Geography*, 19(2), 366-374.

Lamb, W.G.P., & Hunt, J.M. (2000). Multiple Encounter Avoidance Manoeuvres. *Journal of Navigation*, 53(1), 181-186.

Lee, S.M., & Joh, K.Y. (2004). A Fuzzy Logic for Autonomous Navigation of Marine Vehicles Satisfying Colregs Guidelines. *International Journal of Control Automation and Systems*, 2(2), 171-181.

Lenart, A.S. (1983). Collision Threat Parameters for a New Radar Display and Plot Technique. *Journal of Navigation*, 36(3), 404-410.

Lenart, A.S. (1999). Manoeuvring to Required Approach Parameters - CPA Distance and Time. *Annual of Navigation*, 1, 99–108.

Lenart, A.S. (2000). Manoeuvring to Required Approach Parameters - Distance and Time Abeam. *Annual of Navigation*, 2, 81–88.

Li, F., & Hu, S. (2001). Estimates About DCPA and Scope of Alteration of Course of Anti-Collision Action(in Chinese). *Journal of Shanghai Maritime University*, 22(2), 29-32.

Li, S., Meng, Q., & Qu, X. (2012). An Overview of Maritime Waterway Quantitative Risk Assessment Models. *Risk Analysis*, 32(3), 496--512.

Li, Y., & Jin, Y. (2007). MAS and AIS Based Vessel Automatic Collision Avoidance System. *International Conference on Transportation Engineering (ICTE)*, Chengdu, China, 530-530.

Lisowski, J., & Smierzchalski, R. (1995). Assigning of Safe and Optimal Trajectory Avoiding Collision at Sea. *3rd IFAC Workshop Control Applications in Marine System*, Trondheim, 346-350.

Liu, D., Wu, Z., & Jia, C. (2004). Decision-Making Model of D_{cpa}, T_{cpa} and Object's Movement Parameter (in Chinese). *Journal of Dalian Maritime Univiersity*, 30(1).

Liu, Y., & Shi, C. (2005). A Fuzzy-Neural Inference Network for Ship Collision Avoidance. *3rd International Conference on Machine Learning and Cybernetics*, Guangzhou, China.

Lokuge, P., & Alahakoon, D. (2007). Improving the Adaptability in Automated Vessel Scheduling in Container Ports Using Intelligent Software Agents. *European Journal of Operational Research*, 177(3), 1985-2015.

Lomnicki, Z. (1965). A" Branch-and-Bound" Algorithm for the Exact Solution of the Three-Machine Scheduling Problem. *Operational Research Quarterly*, 16(1), 89-100.

Lun, Y.V., Lai, K.-h., & Cheng, T.E. (2010). *Shipping and Logistics Management*, Springer.

Macduff, T. (1974). The Probability of Vessel Collisions. *Ocean Industry*, 9(9), 144-148.

Manley, P. (2002). Collision Regulations-Discussion. *Journal of Navigation*, 55(1), 145-155.

McGeoch, I. (2005). Assessing Risk of Collision. *Journal of Navigation*, 59(1), 169-172.

Menon, R., Sweriduk, G., & Sridhar, B. (1999). Optimal Strategies for Free-Flight Air Traffic Conflict Resolution. *Journal of Guidance Control and Dynamics*, 22, 202-211.

Merrick, J.R.W., Dinesh, V., Singh, A., van Dorp, J.R., & Mazzuchi, T.A. (2003). Propagation of Uncertainty in a Simulation-Based Maritime Risk Assessment Model Utilizing Bayesian Simulation Techniques. *Proceedings of the 2003 Winter Simulation Conference*, 449-455.

Merrick, J.R.W., van Dorp, J.R., Mazzuchi, T., Harrald, J.R., Spahn, J.E., & Grabowski, M. (2002). The Prince William Sound Risk Assessment. *Interfaces*, 32(6), 25-40.

Millns, A.T.C. (1998). Fast Craft and the Colregs. *Journal of Navigation*, 51(3), 430-444.

Montewka, J., Hinz, T., Kujala, P., & Matusiak, J. (2010). Probability Modelling of Vessel Collisions. *Reliability Engineering & System Safety*, 95(5), 573-589.

Morel, G., & Chauvin, C. (2006). A Socio-Technical Approach of Risk Management Applied to Collisions Involving Fishing Vessels. *Safety Science*, 44(7), 599-619.

MPA (1998). Pilotage. Port Marine Circular. Singapore.

MPA (2012). Revision of Anchorages, Fairways and Channels in Port. Port Marine Circular. Singapore.

Ng, W.C., & Wong, C.S. (2006). Evaluating the Impact of Vessel-Traffic Interference on Container Terminal Capacity. *Journal of waterway, port, coastal, and ocean engineering*, 132(2), 76-82.

Ng, W.K., Lim, E.P., Hsu, W.J., & Cao, Y.Y. (1998). Design and Implementation of a Maritime Conflict Prediction System. *The International Electronic Chart Display and Information System Conference (ECDIS)*, Singapore.

Pallottino, L., Feron, E.M., & Bicchi, A. (2002). Conflict Resolution Problems for Air Traffic Management Systems Solved with Mixed Integer Programming. *IEEE Transactions on Intelligent Transportation Systems*, 3(1), 3-11.

Pallottino, L., Scordio, V.G., Bicchi, A., & Frazzoli, E. (2007). Decentralized Cooperative Policy for Conflict Resolution in Multivehicle Systems. *IEEE Transactions on Robotics*, 23(6), 1170-1183.

Pedersen, E., Inoue, K., & Tsugane, M. (2003). Simulator Studies on a Collision Avoidance Display That Facilitates Efficient and Precise Assessment of Evasive Manoeuvres in Congested Waterways. *Journal of Navigation*, 56(3), 411-427.

Pietrzykowski, Z. (1999). Ship Fuzzy Domain in Assessment of Navigational Safety in Restricted Areas. *3rd Navigational Symposium*, Gdynia, 253-264.

Pietrzykowski, Z. (2008). Ship's Fuzzy Domain - a Criterion for Navigational Safety in Narrow Fairways. *Journal of Navigation*, 61(3), 499-514.

Pietrzykowski, Z., & Gucma, L. (2002). Application of the Probabilistic-Fuzzy Method for Assessment of Dangerous Situation of a Ship Manoeuvring in a Restricted Area. *Annual of Navigation*, 4, 61–72.

Prähofer, H., & Kim, T.G. (2000). *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*.

Ronza, A., Félez, S., Darbra, R., Carol, S., Vílchez, J., & Casal, J. (2003). Predicting the Frequency of Accidents in Port Areas by Developing Event Trees from Historical Analysis. *Journal of Loss Prevention in the Process Industries*, 16(6), 551-560.

Salinas, C.F. (2002). Collision Regulation Discussions. *Journal of Navagation*, 55(3), 495-505.

Shi, C., Zhang, M., & Peng, J. (2007). Vessel Collision Avoidance in Close-Quarter Situation Using Differential Games. *International Conference on Transportation Engineering (ICTE)*, Chengdu, China, 277-277.

Slattery, R., Green, S., United States. National, A., Space, A., & Ames Research, C. (1994). Conflict-Free Trajectory Planning for Air Traffic Control Automation.

Smierzchalski, R., & Michalewicz, Z. (1998). Adaptive Modeling of a Ship Trajectory in Collision Situations at Sea. *IEEE International Conference on Evolutionary Computation*, 342-347.

Smierzchalski, R., & Michalewicz, Z. (2000). Modeling of Ship Trajectory in Collision Situations by an Evolutionary Algorithm. *IEEE Transactions on Evolutionary Computation*, 4(3), 227-241.

Stitt, I.P.A. (2003). The Use of VHF in Collision Avoidance at Sea. *Journal of Navigation*, 56(1), 67-78.

Stitt, I.P.A. (2004). AIS and Collision Avoidance - a Sense of Deja Vu. *Journal of Navigation*, 57(2), 167-180.

Sun, L. (2000). The Research on Mathematic Models of Decision-Making in Ship Collision Avoidance (in Chinese). Dalian, Dalian Maritime University. Ph.D.**:** 44-45.

Sunday, D. (2004). "Distance between Lines and Segments with Their Closest Point of Approach." from http://geometryalgorithms.com/Archive/algorithm_0106.

Szlapczynska, J. (2005). Proposal of Multi Criteria Decision Process of Vessel Route Selection for a VTS System. *Zeszyty Naukowe. Transport*, 59, 425-432.

Szlapczynska, J. (2006). Multi Criteria Route Selection with Custom Domain Shape. *Annual of Navigation*, 11, 123-136.

Szlapczynski, R. (2005). A New Method of Ship Routing on Raster Grids, with Turn Penalties and Collision Avoidance. *Journal of Navigation*, 59(1), 27-42.

Szlapczynski, R. (2006). A Unified Measure of Collision Risk Derived from the Concept of a Ship Domain. *Journal of Navigation*, 59(3), 477-490.

Tam, C.K., Bucknall, R., & Greig, A. (2009). Review of Collision Avoidance and Path Planning Methods for Ships in Close Range Encounters. *Journal of Navigation*, 62(3), 455-476.

Tan, B., & Otay, E.N. (1999). Modeling and Analysis of Vessel Casualties Resulting from Tanker Traffic through Narrow Waterways. *Naval Research Logistics (NRL)*, 46(8), 871-892.

Taylor, D.H. (1998). Rules and Regulations in Maritime Collision Avoidance: New Directions for Bridge Team Training. *Journal of Navigation*, 51(1), 67-72.

Tomlin, C., Pappas, G.J., & Sastry, S. (1998). Conflict Resolution for Air Traffic Management: A Study in Multiagent Hybrid Systems. *IEEE Transactions on Automatic Control*, 43(4), 509-521.

Trucco, P., Cagno, E., Ruggeri, F., & Grande, O. (2008). A Bayesian Belief Network Modelling of Organisational Factors in Risk Analysis: A Case Study in Maritime Transportation. *Reliability Engineering & System Safety*, 93(6), 845-856.

Ulusçu, Ö.S., & Altiok, T. (2011). Waiting Time Approximation in Multi-Class Queueing Systems with Multiple Types of Class-Dependent Interruptions. *Annals of Operations Research*, 1-11.

Ulusçu, Ö.S., Özbaş, B., Altıok, T., Or, İ., & Yılmaz, T. (2009). Transit Vessel Scheduling in the Strait of Istanbul. *Journal of Navigation*, 62(1), 59-77.

UNCTAD (2011). *Review of Maritime Transport 2011*.

van Hilten, M.J., & Wolkenfelt, P.H.M. (2000). The Rate of Turn Required for Geographically Fixed Turns: A Formula and Fast-Time Simulations. *Journal of Navigation*, 53(1), 146-155.

Wang, J., Sii, H.S., Yang, J.B., Pillay, A., Yu, D., Liu, J., Maistralis, E., & Saajedi, A. (2004). Use of Advances in Technology for Maritime Risk Assessment. *Risk Analysis: An International Journal*, 24(4), 1041-1063.

Wang, N. (2010). An Intelligent Spatial Collision Risk Based on the Quaternion Ship Domain. *Journal of Navigation*, 63(4), 733-749.

Weber, H. (1998). Fast Craft and the Colregs. *Journal of Navigation*, 51(1), 132-140.

Weng, J., Meng, Q., & Qu, X. (2012). Vessel Collision Frequency Estimation in the Singapore Strait. *Journal of Navigation*, 65(2), 207-221.

Williams, D.L. (1988). *Sea Traffic Management*. London, Ian Allan Ltd.

Wu, Z., & Zheng, Z. (2000). Model of Ship's Optimization Opportunity for Taking Action in Decision-Making for Collision Avoidance (in Chinese). *Journal of Dalian Maritime Univiersity*, 26(4).

Yao, J., Ren, Y., & Li, Y. (2010). Research of Marine Accident Based on FTA Method. *Journal of Dalian Ocean University*, 25(4), 348-352.

Yap, W.Y., & Lam, J.S.L. (2013). 80 Million-Twenty-Foot-Equivalent-Unit Container Port? Sustainability Issues in Port and Coastal Development. *Ocean & Coastal Management*.

Zec, D. (1996). An Algorithm for a Real-Time Detection of Encounter Situations. *Journal of Navigation*, 49(1), 121.

Zeigler, B.P., Praehofer, H., & Kim, T.G. (1976). *Theory of Modeling and Simulation*, John Wiley New York.

Zhang, G., Sheng, J., Zhang, G., & Wang, X. (2009). Research on Collision-Avoidance System of Ship with AIS Based on APT. *The Eighth International Conference of Chinese Logistices and Transportation Professionals*, 1295-1300.

Zhao, J., & Wang, F. (1989). Model for Determine Right Movement for Collision Avoidance Action. *Journal of Dalian Maritime Univiersity*, 15(3).

Zhao, J., & Wang, F. (1991). Simulation Model of DCPA Decision Making. *7th International Congress of IAIN*, Cairo.

Zhao, J., Wu, Z., & Wang, F. (1993). Comments of Ship Domains. *Journal of Navigation*, 46(3), 422-436.

Zhu, J.Q. (2003). Probabilistic Conflict Risk Model for Marine Traffic within a Sea Port. Nanyang Technological University. Master of Science.

Zhu, X.L., Xu, H.Z., & Lin, J.Q. (2001). Domain and Its Model Based on Neural Networks. *Journal of Navigation*, 54(1), 97-103.

Zhu, Z., & Liu, W. (2004). Research on Collision-Avoidance System of Ship Based on Geography Information System (in Chinese). *Ship Engineering*, 26(1), 42-45.

Zsambok, C.E., & Klein, G.A. (1997). *Naturalistic Decision Making*, Lawrence Erlbaum.

Appendix A
Relationship between two domains

This section introduces our method to check how to evaluate the relationship between two domains, which are rectangles. Moreover, the proposed method can also be directly used to check the relationship between two parallelograms. Thus, the method presented here is used in the both determining a conflict and predicting a conflict.

Two functions

In this section, two functions, which can be applied to evaluate the relationship between two rectangles, are introduced.

Function 1. $L_{side}\left(a,b,c\right)$

$L_{side}\left(a,b,c\right)$ can be used to identify the position of a point relative to a directed line segment. In the two-dimensional plane shown in Figure A.1, c is a point and $\overrightarrow{ab}$ is a directed line segment.

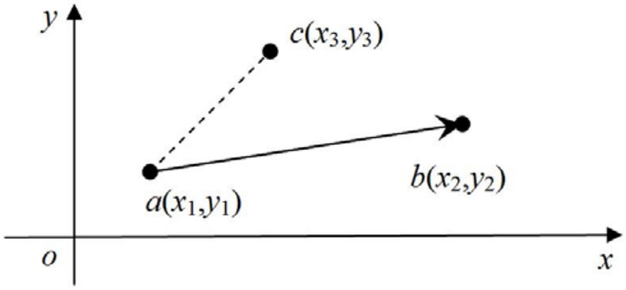


Figure A.1 A point and a line segment.

The cross product and dot product on the plane are defined as

$$(x_1, y_1) \otimes (x_2, y_2) = x_1 y_2 - x_2 y_1,$$
$$(x_1, y_1) \cdot (x_2, y_2) = x_1 x_2 - y_1 y_2.$$

Facing the direction from *a* to *b*, it can be determined that the point *c* is on which side of the line segment $\overrightarrow{ab}$. Let

$$L_{side}\left(a,b,c\right)=\left(b-a\right)\otimes\left(c-a\right). \tag{A.1}$$

Then, the following conclusions can be obtained:

- When $L_{side}\left(a,b,c\right)>0$, c is on the left-hand side of $\overrightarrow{ab}$;
- When $L_{side}\left(a,b,c\right)<0$, c is on the right-hand side of $\overrightarrow{ab}$;
- Otherwise, $L_{side}\left(a,b,c\right)=0$, *c* is on the line $\overrightarrow{ab}$.

Function 2: $L_{\text{intersect}}\left(a,b,c,d\right)$

$L_{\text{inersect}}\left(a,b,c,d\right)$ can be used to identify whether two line segments intersect. Suppose $\overrightarrow{ab}$ and $\overrightarrow{cd}$ are two line segments in a two-dimensional plane (Figure A.2).

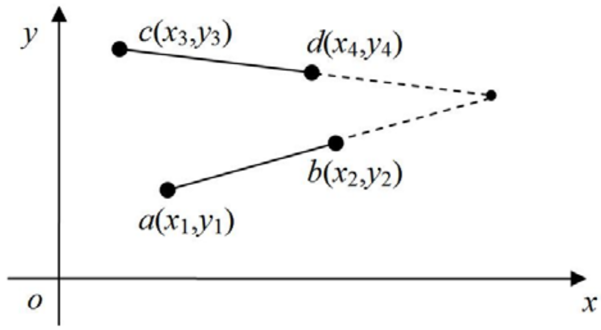


Figure A.2 Two line segments.

If the point *a* and the point *b* are located on different sides of the line segment $\overrightarrow{cd}$, while the point *c* and the point *d* are located on different sides of the line segment $\overrightarrow{ab}$, the two line segments $\overrightarrow{ab}$ and $\overrightarrow{cd}$ intersect. Based on this criterion, it can be identified whether the two line segments intersect. Let

$$\begin{aligned}L_{\text{inersect}}\left(a,b,c,d\right)&=\left[\left(\alpha_1>0,\alpha_2<0\right)\vee\left(\alpha_1<0,\alpha_2>0\right)\right]\\&\quad\wedge\left[\left(\alpha_3>0,\alpha_4<0\right)\vee\left(\alpha_3<0,\alpha_4>0\right)\right]\\&=\left(\alpha_1\alpha_2<0\right)\wedge\left(\alpha_3\alpha_4<0\right),\end{aligned} \tag{A.2}$$

where

$$\begin{aligned}\alpha_1&=L_{side}\left(a,b,c\right),\quad \alpha_2=L_{side}\left(a,b,d\right),\\ \alpha_3&=L_{side}\left(c,d,a\right),\quad \alpha_4=L_{side}\left(c,d,b\right).\end{aligned}$$

and $\vee$ means ‘logical or’ , $\wedge$ means ‘logical and’.

The value for $L_{\text{inersect}}\left(a,b,c,d\right)$ should be ‘TRUE’ or ‘FALSE’. When $L_{\text{inersect}}\left(a,b,c,d\right)$ is ‘TRUE’, the line segment $\overrightarrow{ab}$ intersects with the line segment $\overrightarrow{cd}$.

With reference to Figure A.3, facing the direction D_1, denote the vertex in the lower right-hand corner of Q_1 as p_1^1 and moving in counterclockwise direction, denote the other three vertices as p_1^2, p_1^3, p_1^4. Similarly, the four vertices of $Q_2(p_2^1, p_2^2, p_2^3, p_2^4)$ are denoted counterclockwise from p_2^1 . Now, designate the line segment $\overrightarrow{p_1^1 p_1^2}$ as the first edge of Q_1, and then in turn designate $\overrightarrow{p_1^2 p_1^3}$, $\overrightarrow{p_1^3 p_1^4}$ and $\overrightarrow{p_1^4 p_1^1}$ as the second, third and forth edges, respectively. The same process is followed for Q_2. The four edges of $Q_2(\overrightarrow{p_2^1 p_2^2}, \overrightarrow{p_2^2 p_2^3}, \overrightarrow{p_2^3 p_2^4}, \overrightarrow{p_2^4 p_2^1})$ are designated counterclockwise from $\overrightarrow{p_2^1 p_2^2}$.

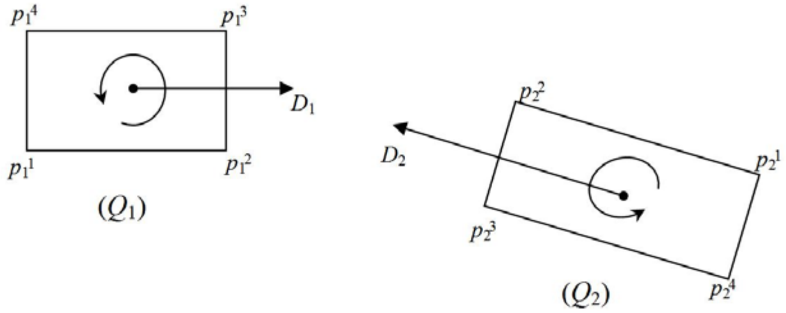


Figure A.3 Q_1 and Q_2.

Hence, the spatial positions of the eight vertices can be obtained,

$$
\begin{aligned}
&p_1^1\left(x_1-\frac{L_1}{2}-\overline{L_1^2},\, y_1-\frac{W_1}{2}-\overline{W_1}\right), \quad && p_1^2\left(x_1+\frac{L_1}{2}+\overline{L_1^1},\, y_1-\frac{W_1}{2}-\overline{W_1}\right),\\
&p_1^3\left(x_1+\frac{L_1}{2}+\overline{L_1^1},\, y_1+\frac{W_1}{2}+\overline{W_1}\right), \quad && p_1^4\left(x_1-\frac{L_1}{2}-\overline{L_1^1},\, y_1+\frac{W_1}{2}+\overline{W_1}\right),\\
&p_2^1\left(x_2-\frac{L_2}{2}-\overline{L_2^2},\, y_2-\frac{W_2}{2}-\overline{W_2}\right), \quad && p_2^2\left(x_2+\frac{L_2}{2}+\overline{L_2^1},\, y_2-\frac{W_2}{2}-\overline{W_2}\right),\\
&p_2^3\left(x_2+\frac{L_2}{2}+\overline{L_2^1},\, y_2+\frac{W_2}{2}+\overline{W_2}\right), \quad && p_2^4\left(x_2-\frac{L_2}{2}-\overline{L_2^1},\, y_2+\frac{W_2}{2}+\overline{W_2}\right),
\end{aligned}
$$

where

x_i, y_i : the coordinates of the center point of vessel V_i ,

L_i, W_i : the length and width of vessel V_i ,

$\overline{L_i^1}$: the longitudinal clearance in the direction of the bow for vessel V_i ,

$\overline{L_i^2}$: the longitudinal clearance in the direction of the stern for vessel V_i ,

$\overline{W_i}$: the lateral clearance of vessel V_i .

Let

$$\alpha_{ij}^k = L_{side}(p_i^j, p_i^{j+1}, p_{i+1}^k), \quad i=1,2, \quad j,k=1,2,3,4, \tag{A.3}$$

where p_i^j is the j -th vertex of the i -th rectangle, and $p_i^5 = p_i^1, p_3^k = p_1^k$.

Eq.(A.3) is used to determine which side of line segment $\overrightarrow{p_i^j p_i^{j+1}}$ where the vertex p_{i+1}^k locates. For example, $\alpha_{24}^4 = L_{side}(p_2^4, p_2^5, p_3^4) = L_{side}(p_2^4, p_2^1, p_1^4)$ is used to identify the vertex p_1^4 (of Q_1) is on which side of the edge $\overrightarrow{p_2^4 p_2^1}$ (of Q_2). Set

$$\begin{aligned} \beta_{ij}^k &= L_{intersect}(p_i^j, p_i^{j+1}, p_{i+1}^k, p_{i+1}^{k+1}) \\ &= \left(\alpha_{ij}^k \alpha_{ij}^{k+1} < 0\right) \wedge \left(\alpha_{i+1,k}^j \alpha_{i+1,k}^{j+1} < 0\right), \\ i &= 1,2, \quad j,k=1,2,3,4, \end{aligned} \tag{A.4}$$

where $\alpha_{ij}^5 = \alpha_{ij}^1, \alpha_{3k}^j = \alpha_{1k}^j$.

Eq.(A.4) is used to identify whether the two line segments $\overrightarrow{p_i^j p_i^{j+1}}$ and $\overrightarrow{p_{i+1}^k p_{i+1}^{k+1}}$ intersect. The value for β_{ij}^k should be 'TRUE' or 'FALSE'. For example,

$$\beta_{24}^4 = \left(\alpha_{24}^4 \alpha_{24}^5 < 0\right) \wedge \left(\alpha_{34}^4 \alpha_{34}^5 < 0\right) = \left(\alpha_{24}^4 \alpha_{24}^1 < 0\right) \wedge \left(\alpha_{14}^4 \alpha_{14}^1 < 0\right).$$

It is used to identify whether the edge $\overrightarrow{p_2^4 p_2^1}$ (of Q_2) and the edge $\overrightarrow{p_1^4 p_1^1}$ (of Q_1) intersect.

Check whether Q_1 and Q_2 intersect

Following the flowchart in Figure 4.11, the algorithm will first check whether the two domains intersect. Eq.(A.4)can be used to identify whether Q_1 and Q_2 intersect. If any edge of Q_1 intersects with any edge of Q_2, the two rectangles intersect with each other. Each time, the algorithm randomly selects two edges, each from Q_1 and Q_2, and evaluate whether they intersect. If yes, it implies that Q_1 and Q_2 intersect. Otherwise, the evaluation will continue until all edge pairs have been evaluated. For two rectangles each with four edges, the algorithm needs such evaluation for $4 \times 4 = 16$ times.

Check whether Q_1 is enclosed within Q_2

Following the flowchart in Figure 4.11, if the two domains do not intersect, the algorithm needs to check whether one is enclosed by another. Eq.(A.3)is used to identify whether Q_1 is enclosed within Q_2. If all vertices of Q_1 are on the left-hand side of the four edges of Q_2, that is to say p_1^1, p_1^2, p_1^3, and p_1^4 are all on the left-hand side of $\overrightarrow{p_2^1 p_2^2}$, $\overrightarrow{p_2^2 p_2^3}$, $\overrightarrow{p_2^3 p_2^4}$ and $\overrightarrow{p_2^4 p_2^1}$, Q_1 is enclosed within Q_2.

There are rare cases when a rectangle that is enclosed in another rectangle is rare. Thus, only when a vertex of Q_1 is not on the left-hand side of an edge of Q_2, e.g. p_1^1 is on the left-hand side of $\overrightarrow{p_2^1 p_2^2}$, $\overrightarrow{p_2^2 p_2^3}$, $\overrightarrow{p_2^3 p_2^4}$ but is on the right-hand side of $\overrightarrow{p_2^4 p_2^1}$, Q_1, it can be concluded that Q_1 is not enclosed in Q_2. Each time, the algorithm randomly selects an edge of Q_1 and evaluate whether $\alpha_{ij}^k \leq 0$ (Eq.(A.3)). If yes, it means that Q_1 is not enclosed in Q_2, and the evaluation ends. Otherwise, the evaluation will continue.

Check whether Q_2 is enclosed within Q_1

A similar procedure is used to identify whether Q_2 is enclosed within Q_1. Each time, the algorithm randomly selects a vertex of Q_2 and evaluates whether $\alpha_{ij}^k \leq 0$. If yes, it means that Q_2 is not enclosed in Q_1. Otherwise, the evaluation will continue.

Appendix B
Traffic network statistics

This section lists the statistics of the traffic network of the Port of Singapore.

Table B.1 The node with its regions.

ID	Type	Regions	ID	Type	Regions	ID	Type	Regions
1	berth	58	24	anchorage	1	73	berth	1
2	berth	42	25	boarding	2	74	berth	2
3	berth	27	26	anchorage	1	75	berth	13
4	berth	43	27	anchorage	1	76	berth	11
5	berth	36	28	anchorage	1	77	anchorage	1
6	berth	22	29	anchorage	1	78	berth	24
7	berth	27	30	boarding	1	79	berth	7
8	berth	10	31	boarding	2	80	berth	18
9	berth	13	32	anchorage	1	81	anchorage	1
10	berth	9	33	anchorage	1	83	anchorage	1
11	berth	8	34	boarding	1	84	boarding	2
12	berth	8	35	boarding	2	94	anchorage	1
13	berth	7	60	anchorage	2	96	berth	1
14	berth	9	63	anchorage	1	98	berth	1
15	boarding	2	64	anchorage	1	100	berth	2
16	boarding	1	65	anchorage	4	102	berth	10
17	berth	1	66	anchorage	1	104	berth	8
18	berth	1	67	anchorage	1	106	berth	27
19	boarding	2	68	anchorage	1	108	berth	4
20	anchorage	1	69	anchorage	2	110	anchorage	1
21	anchorage	7	70	anchorage	1	112	berth	1
22	anchorage	2	71	anchorage	2	115	berth	2
23	anchorage	2	72	anchorage	1	117	berth	2

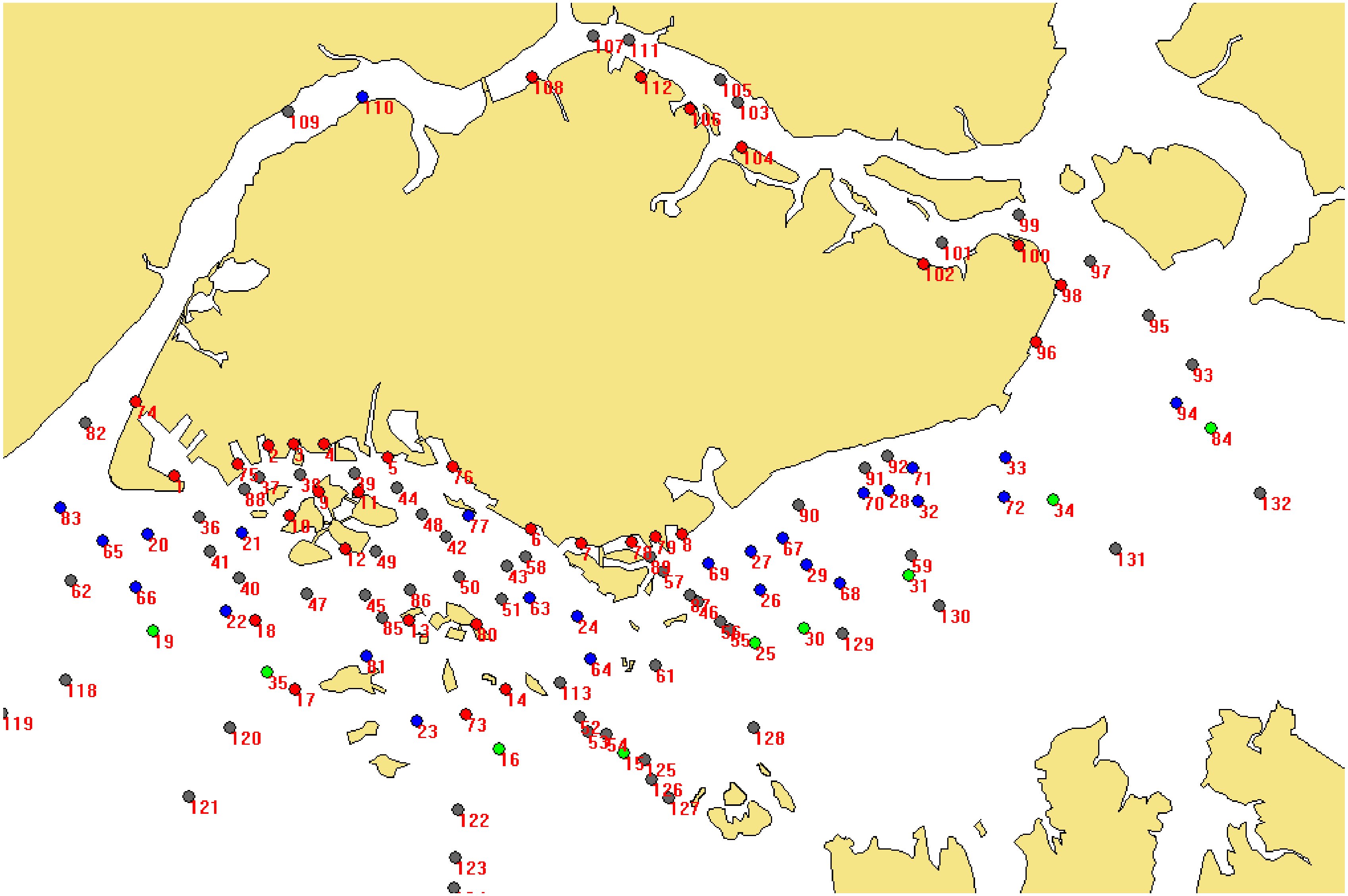


Figure B.1 The node with its ID.

Table B.2 List of nodes.

ID	x	y	type	ID	x	y	type	ID	x	y	type	ID	x	y	type
1	131	389	berth	34	794	407	boarding	67	590	436	anchorage	100	768	215	berth
2	202	366	berth	35	201	537	boarding	68	633	470	anchorage	101	710	213	intersection
3	221	365	berth	36	150	420	intersection	69	534	455	anchorage	102	696	229	berth
4	244	365	berth	37	195	390	intersection	70	651	402	anchorage	103	556	107	intersection
5	292	375	berth	38	226	388	intersection	71	688	383	anchorage	104	559	141	berth
6	400	429	berth	39	267	387	intersection	72	757	405	anchorage	105	543	90	intersection
7	438	440	berth	40	180	466	intersection	73	351	569	berth	106	520	112	berth
8	514	433	berth	41	158	446	intersection	74	102	333	berth	107	447	57	intersection
9	240	401	berth	42	336	435	intersection	75	179	380	berth	108	401	88	berth
10	218	419	berth	43	382	457	intersection	76	341	382	berth	109	217	114	intersection
11	270	401	berth	44	299	398	intersection	77	353	419	anchorage	110	273	103	anchorage
12	260	444	berth	45	275	479	intersection	78	476	439	berth	111	474	60	intersection
13	308	498	berth	46	526	484	intersection	79	494	435	berth	112	483	88	berth
14	381	550	berth	47	231	478	intersection	80	359	501	berth	113	422	545	intersection
15	470	598	boarding	48	318	418	intersection	81	276	525	anchorage	114	-140	415	intersection
16	376	595	boarding	49	283	446	intersection	82	64	349	intersection	115	-167	500	berth
17	222	550	berth	50	346	465	intersection	83	45	413	anchorage	116	-160	455	intersection
18	192	498	berth	51	378	482	intersection	84	913	353	boarding	117	1498	355	berth
19	115	506	boarding	52	437	571	intersection	85	288	496	intersection	118	49	543	intersection
20	111	433	anchorage	53	443	582	intersection	86	309	475	intersection	119	1	568	intersection
21	182	432	anchorage	54	457	584	intersection	87	520	479	intersection	120	173	579	intersection
22	170	491	anchorage	55	550	505	intersection	88	184	399	intersection	121	142	631	intersection
23	314	574	anchorage	56	543	499	intersection	89	490	450	intersection	122	345	641	intersection
24	435	495	anchorage	57	500	461	intersection	90	602	411	intersection	123	343	677	intersection
25	569	515	boarding	58	396	450	intersection	91	652	383	intersection	124	342	700	intersection
26	573	475	anchorage	59	687	449	intersection	92	669	374	intersection	125	486	603	intersection
27	566	446	anchorage	60	342	730	anchorage	93	899	305	intersection	126	491	618	intersection
28	670	400	anchorage	61	494	532	intersection	94	887	334	anchorage	127	504	632	intersection
29	608	456	anchorage	62	53	468	intersection	95	866	268	intersection	128	568	579	intersection
30	606	504	boarding	63	399	481	anchorage	96	781	288	berth	129	635	508	intersection
31	685	464	boarding	64	445	527	anchorage	97	822	227	intersection	130	708	487	intersection
32	692	408	anchorage	65	77	438	anchorage	98	800	245	berth	131	841	444	intersection
33	758	375	anchorage	66	102	473	anchorage	99	768	192	intersection	132	950	402	intersection

Note: x and y values are the planar screen coordinates in the system. The unit is pixel.

Table B.3 List of links (part 1).

ID	From	To	Type	Length	ID	From	To	Type	Length	ID	From	To	Type	Length
1	1	36	dual	3150	31	20	62	dual	100	61	36	41	dual	1800
2	2	37	dual	1000	32	20	65	dual	2650	62	37	38	dual	2100
3	3	38	dual	650	33	21	41	dual	800	63	39	44	dual	1450
4	4	39	dual	1250	34	22	35	dual	1200	64	40	41	dual	1850
5	5	44	dual	1550	35	22	47	dual	2300	65	42	43	dual	3300
6	6	58	dual	1700	36	24	58	dual	650	66	42	49	dual	3750
7	7	58	dual	2100	37	24	61	dual	3750	67	43	50	dual	2300
8	8	57	dual	2100	38	25	26	dual	1050	68	43	51	dual	1650
9	9	39	dual	2100	39	25	30	dual	2500	69	43	58	dual	310
10	10	38	dual	2550	40	25	55	dual	1350	70	44	48	dual	1600
11	11	44	dual	1200	41	26	56	dual	1500	71	45	47	dual	2800
12	12	47	dual	3500	42	27	30	dual	2900	72	45	86	single	2160
13	12	49	dual	1700	43	27	69	dual	300	73	49	45	single	2250
14	13	50	dual	1200	44	28	59	dual	1950	74	46	56	dual	1400
15	14	15	dual	3000	45	28	91	dual	500	75	40	47	dual	3200
16	14	52	dual	700	46	29	30	dual	1500	76	42	48	dual	1200
17	15	16	dual	5700	47	29	59	dual	3100	77	51	113	dual	2400
18	15	25	dual	9000	48	30	26	dual	1350	78	52	53	dual	800
19	15	54	dual	1150	49	30	31	dual	5600	79	53	54	dual	900
20	16	23	dual	2000	50	30	68	dual	1000	80	53	61	dual	4600
21	16	73	dual	3000	51	31	32	dual	2100	81	54	55	dual	8000
22	17	35	dual	900	52	31	34	dual	7500	82	55	56	dual	600
23	18	35	dual	3900	53	31	59	dual	1000	83	61	56	dual	3700
24	18	45	dual	2100	54	33	34	dual	1650	84	63	43	dual	1700
25	18	85	dual	2100	55	33	92	dual	1100	85	63	51	dual	800
26	19	22	dual	1500	56	34	72	dual	1600	86	63	58	dual	600
27	19	35	dual	6000	57	34	84	dual	8100	87	64	52	dual	1500
28	19	40	dual	5300	58	45	35	single	6000	88	64	61	dual	2100
29	19	62	dual	4750	59	35	85	dual	5900	89	65	40	dual	3500
30	20	40	dual	850	60	36	20	dual	1800	90	65	62	dual	800

Table B.4 List of links (part 2).

ID	From	To	Type	Length	ID	From	To	Type	Length	ID	From	To	Type	Length
91	66	19	dual	1140	121	57	90	dual	7000	151	120	118	single	7800
92	66	40	dual	700	122	59	90	dual	5700	152	119	121	single	10700
93	66	62	dual	2800	123	90	91	dual	3600	153	16	122	dual	3800
94	67	59	dual	6000	124	91	92	dual	1250	154	122	120	dual	11050
95	67	90	dual	1150	125	93	94	dual	700	155	122	123	dual	2200
96	68	59	dual	2200	126	93	95	dual	3300	156	121	123	single	13800
97	69	27	dual	300	127	95	96	dual	2850	157	123	124	dual	1800
98	69	57	dual	600	128	95	97	dual	4200	158	60	124	dual	21300
99	70	59	dual	2400	129	97	98	dual	3100	159	121	124	single	14800
100	70	90	dual	1250	130	97	99	dual	4000	160	15	125	dual	1100
101	71	92	dual	550	131	99	100	dual	1100	161	125	122	single	8800
102	74	82	dual	600	132	99	101	dual	4000	162	123	126	single	9600
103	75	88	dual	850	133	101	102	dual	1100	163	124	127	single	11900
104	76	48	dual	2800	134	101	103	dual	16000	164	25	128	dual	4100
105	77	42	dual	3200	135	103	104	dual	700	165	128	125	single	5200
106	78	89	dual	1100	136	103	105	dual	1400	166	126	128	single	5450
107	79	89	dual	950	137	105	106	dual	400	167	127	128	single	5200
108	80	51	dual	700	138	105	111	dual	4900	168	30	129	dual	1800
109	81	85	dual	900	139	107	108	dual	300	169	128	129	dual	5900
110	82	109	dual	21000	140	107	111	dual	1700	170	31	130	dual	2200
111	62	83	dual	3650	141	109	110	dual	1100	171	129	130	dual	4700
112	82	83	dual	4600	142	111	112	dual	450	172	34	131	dual	3600
113	84	93	dual	3200	143	52	113	dual	1900	173	130	131	dual	8400
114	85	86	single	1900	144	113	24	dual	600	174	84	132	dual	4100
115	86	50	single	2400	145	117	132	dual	32900	175	131	132	dual	7300
116	46	87	dual	500	146	19	118	dual	5200	176	125	126	dual	1000
117	57	87	dual	1800	147	118	114	dual	21050	177	126	127	dual	1200
118	36	88	dual	2600	148	119	115	dual	24200	178	120	121	dual	4000
119	37	88	dual	900	149	119	116	dual	24200	179	57	69	dual	1200
120	57	89	dual	900	150	35	120	dual	3300	180	119	118	single	3800

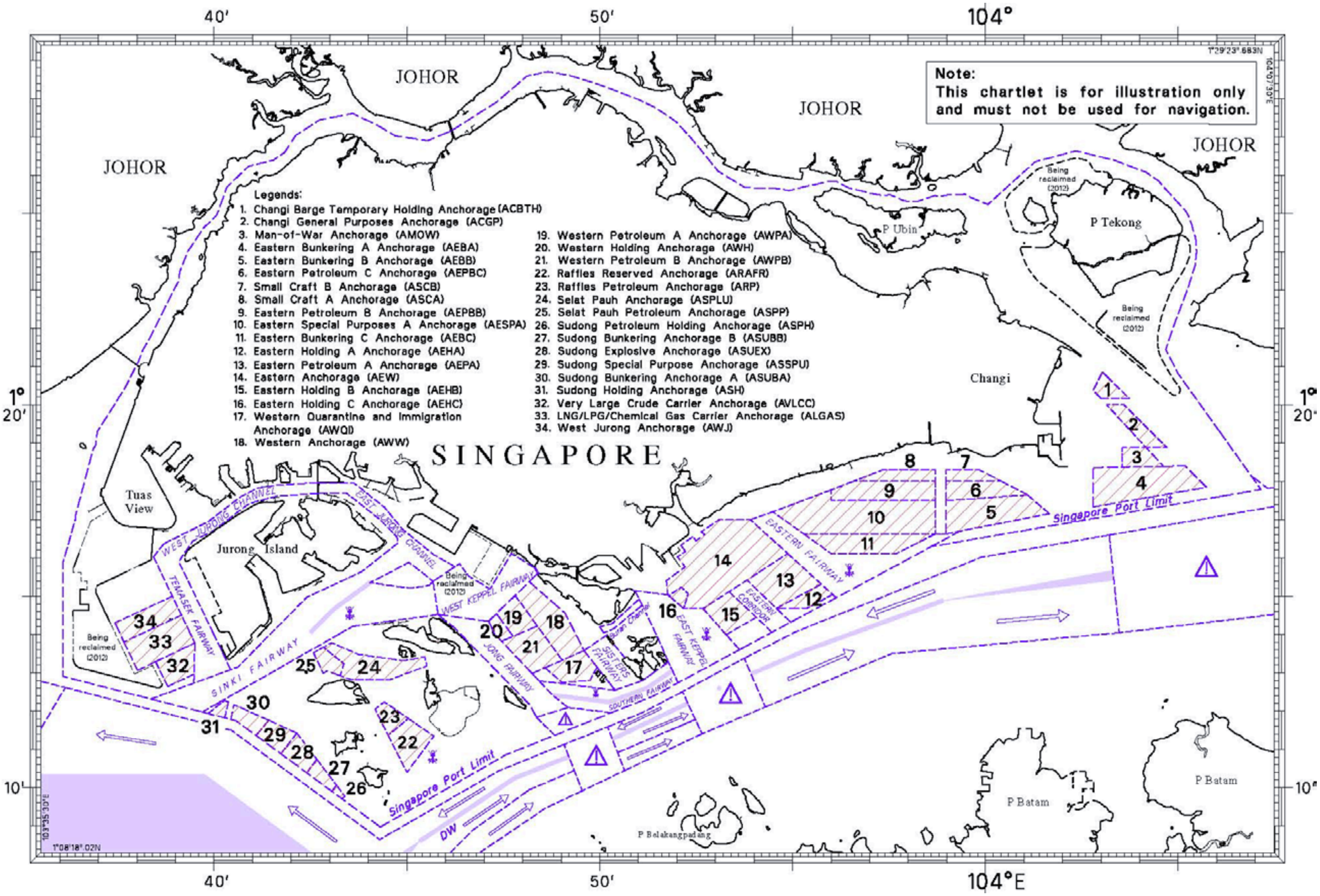


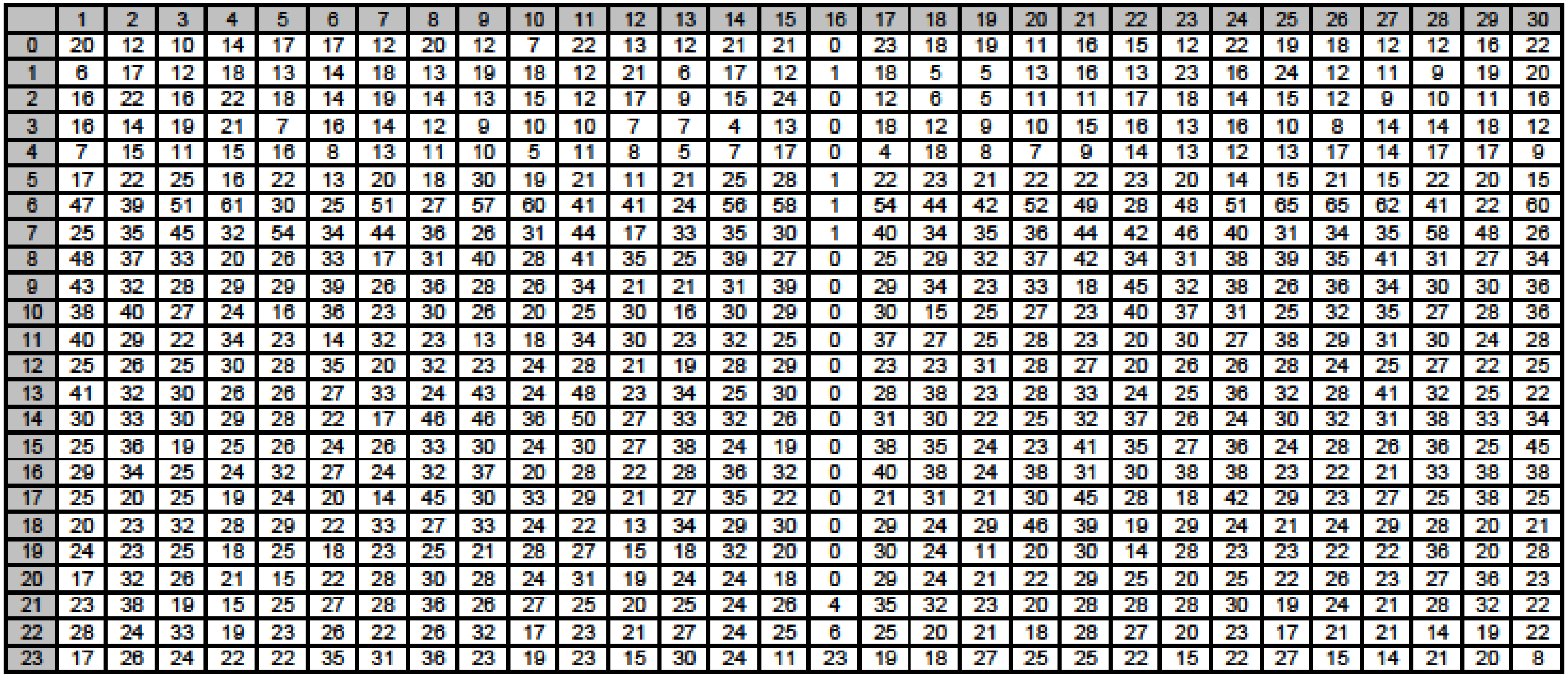
Figure B.2 Port of Singapore – anchorages, fairways and channels.

Appendix C

Vessel arrival statistics

This section lists the schedule data input to our simulation system. The data reflects the arrival information collected for 30 days from the Port of Singapore. The data listed are presented in two forms: number of arrival vessels per hour and number of vessels in the seaport per hour.

Table C.1 Number of arrival vessels per hour in 30 days.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
0	20	12	10	14	17	17	12	20	12	7	22	13	12	21	21	0	23	18	19	11	16	15	12	22	19	18	12	12	16	22
1	6	17	12	18	13	14	18	13	19	18	12	21	6	17	12	1	18	5	5	13	16	13	23	16	24	12	11	9	19	20
2	16	22	16	22	18	14	19	14	13	15	12	17	9	15	24	0	12	6	5	11	11	17	18	14	15	12	9	10	11	16
3	16	14	19	21	7	16	14	12	9	10	10	7	7	4	13	0	18	12	9	10	15	16	13	16	10	8	14	14	18	12
4	7	15	11	15	16	8	13	11	10	5	11	8	5	7	17	0	4	18	8	7	9	14	13	12	13	17	14	17	17	9
5	17	22	25	16	22	13	20	18	30	19	21	11	21	25	28	1	22	23	21	22	22	23	20	14	15	21	15	22	20	15
6	47	39	51	61	30	25	51	27	57	60	41	41	24	56	58	1	54	44	42	52	49	28	48	51	65	65	62	41	22	60
7	25	35	45	32	54	34	44	36	26	31	44	17	33	35	30	1	40	34	35	36	44	42	46	40	31	34	35	58	48	26
8	48	37	33	20	26	33	17	31	40	28	41	35	25	39	27	0	25	29	32	37	42	34	31	38	39	35	41	31	27	34
9	43	32	28	29	29	39	26	36	28	26	34	21	21	31	39	0	29	34	23	33	18	45	32	38	26	36	34	30	30	36
10	38	40	27	24	16	36	23	30	26	20	25	30	16	30	29	0	30	15	25	27	23	40	37	31	25	32	35	27	28	36
11	40	29	22	34	23	14	32	23	13	18	34	30	23	32	25	0	37	27	25	28	23	20	30	27	38	29	31	30	24	28
12	25	26	25	30	28	35	20	32	23	24	28	21	19	28	29	0	23	23	31	28	27	20	26	26	28	24	25	27	22	25
13	41	32	30	26	26	27	33	24	43	24	48	23	34	25	30	0	28	38	23	28	33	24	25	36	32	28	41	32	25	22
14	30	33	30	29	28	22	17	46	46	36	50	27	33	32	26	0	31	30	22	25	32	37	26	24	30	32	31	38	33	34
15	25	36	19	25	26	24	26	33	30	24	30	27	38	24	19	0	38	35	24	23	41	35	27	36	24	28	26	36	25	45
16	29	34	25	24	32	27	24	32	37	20	28	22	28	36	32	0	40	38	24	38	31	30	38	38	23	22	21	33	38	38
17	25	20	25	19	24	20	14	45	30	33	29	21	27	35	22	0	21	31	21	30	45	28	18	42	29	23	27	25	38	25
18	20	23	32	28	29	22	33	27	33	24	22	13	34	29	30	0	29	24	29	46	39	19	29	24	21	24	29	28	20	21
19	24	23	25	18	25	18	23	25	21	28	27	15	18	32	20	0	30	24	11	20	30	14	28	23	23	22	22	36	20	28
20	17	32	26	21	15	22	28	30	28	24	31	19	24	24	18	0	29	24	21	22	29	25	20	25	22	26	23	27	36	23
21	23	38	19	15	25	27	28	36	26	27	25	20	25	24	26	4	35	32	23	20	28	28	28	30	19	24	21	28	32	22
22	28	24	33	19	23	26	22	26	32	17	23	21	27	24	25	6	25	20	21	18	28	27	20	23	17	21	21	14	19	22
23	17	26	24	22	22	35	31	36	23	19	23	15	30	24	11	23	19	18	27	25	25	22	15	22	27	15	14	21	20	8

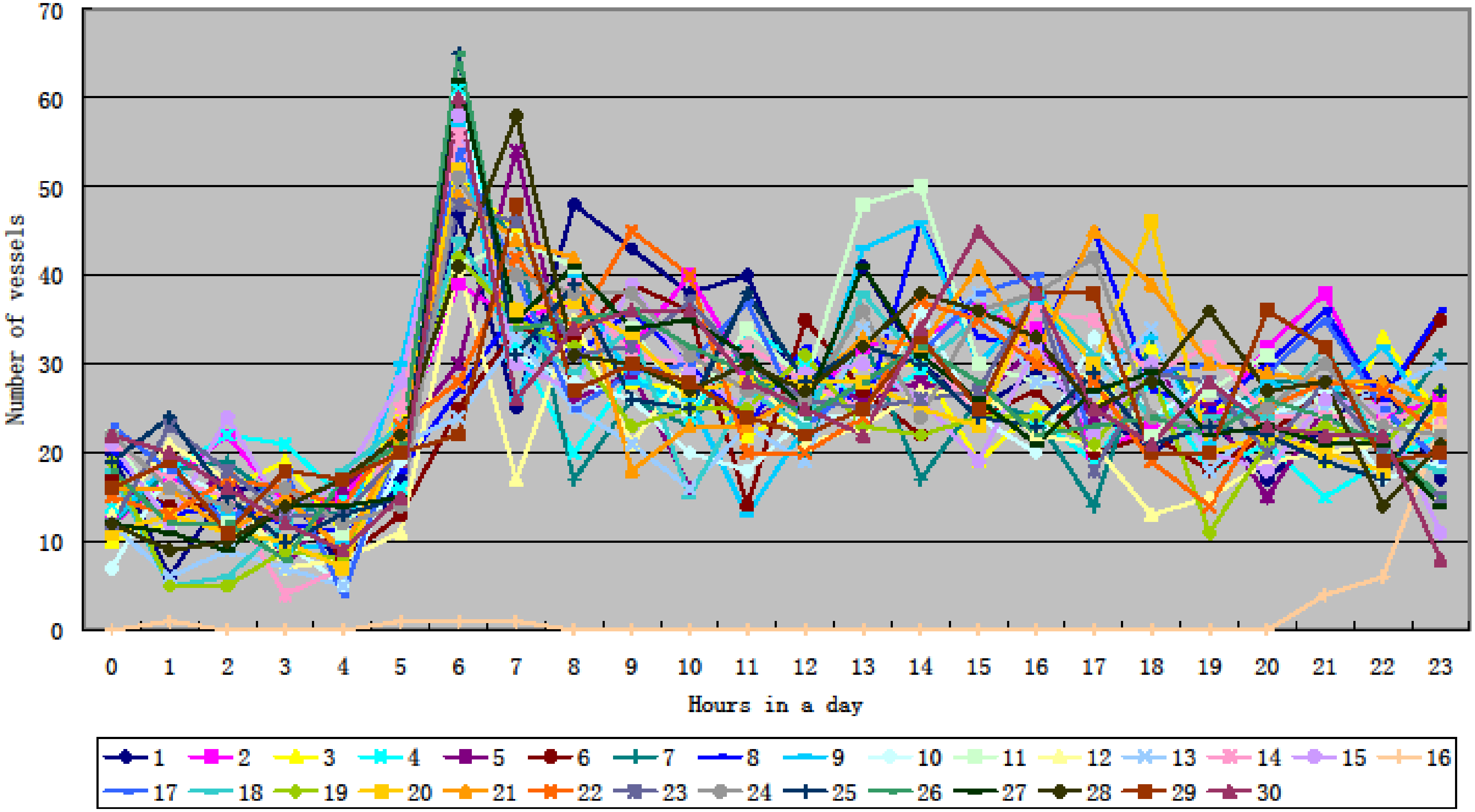
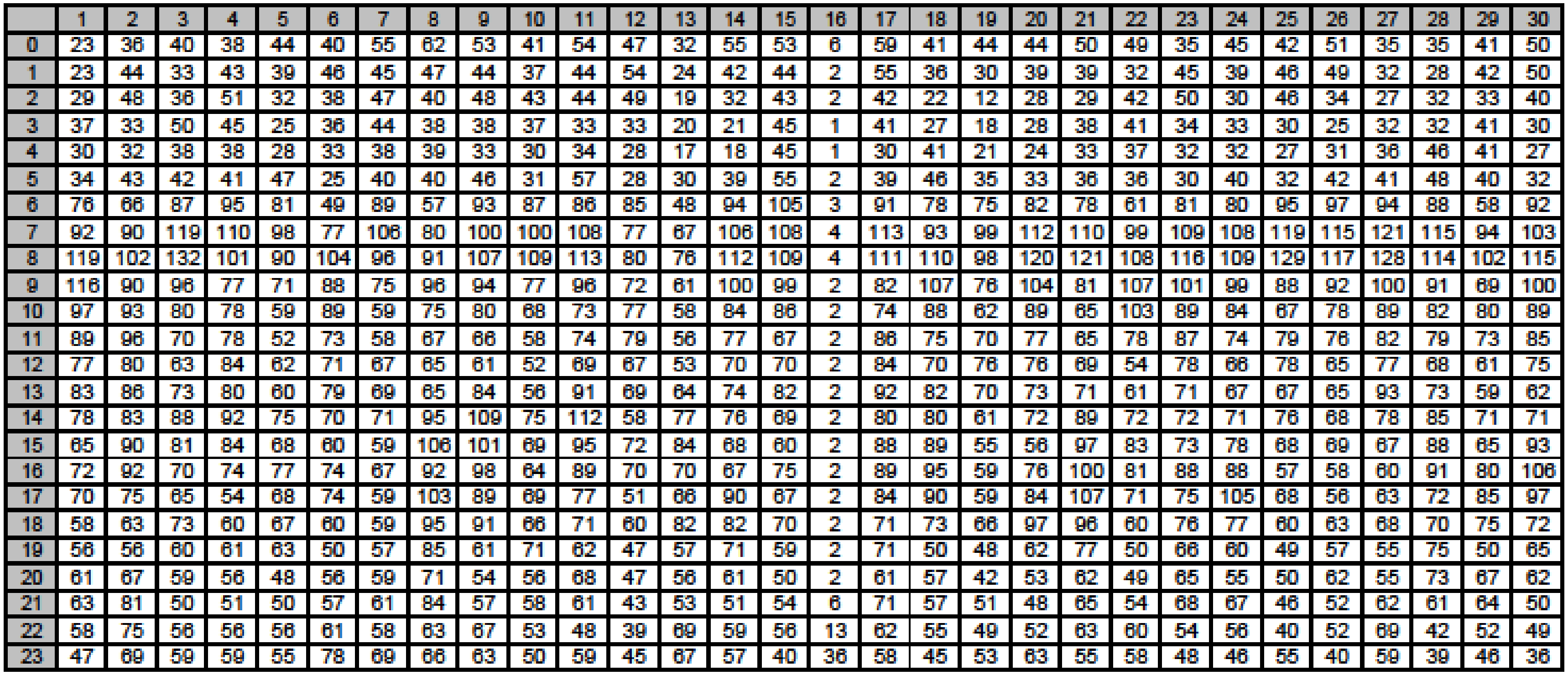


Figure C.1 Number of arrival vessels per hour in 30 days.

Table C.2 Number of vessels in the seaport per hour in 30 days.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
0	23	36	40	38	44	40	55	62	53	41	54	47	32	55	53	6	59	41	44	44	50	49	35	45	42	51	35	35	41	50
1	23	44	33	43	39	46	45	47	44	37	44	54	24	42	44	2	55	36	30	39	39	32	45	39	46	49	32	28	42	50
2	29	48	36	51	32	38	47	40	48	43	44	49	19	32	43	2	42	22	12	28	29	42	50	30	46	34	27	32	33	40
3	37	33	50	45	25	36	44	38	38	37	33	33	20	21	45	1	41	27	18	28	38	41	34	33	30	25	32	32	41	30
4	30	32	38	38	28	33	38	39	33	30	34	28	17	18	45	1	30	41	21	24	33	37	32	32	27	31	36	46	41	27
5	34	43	42	41	47	25	40	40	46	31	57	28	30	39	55	2	39	46	35	33	36	36	30	40	32	42	41	48	40	32
6	76	66	87	95	81	49	89	57	93	87	86	85	48	94	105	3	91	78	75	82	78	61	81	80	95	97	94	88	58	92
7	92	90	119	110	98	77	106	80	100	100	108	77	67	106	108	4	113	93	99	112	110	99	109	108	119	115	121	115	94	103
8	119	102	132	101	90	104	96	91	107	109	113	80	76	112	109	4	111	110	98	120	121	108	116	109	129	117	128	114	102	115
9	116	90	96	77	71	88	75	96	94	77	96	72	61	100	99	2	82	107	76	104	81	107	101	99	88	92	100	91	69	100
10	97	93	80	78	59	89	59	75	80	68	73	77	58	84	86	2	74	88	62	89	65	103	89	84	67	78	89	82	80	89
11	89	96	70	78	52	73	58	67	66	58	74	79	56	77	67	2	86	75	70	77	65	78	87	74	79	76	82	79	73	85
12	77	80	63	84	62	71	67	65	61	52	69	67	53	70	70	2	84	70	76	76	69	54	78	66	78	65	77	68	61	75
13	83	86	73	80	60	79	69	65	84	56	91	69	64	74	82	2	92	82	70	73	71	61	71	67	67	65	93	73	59	62
14	78	83	88	92	75	70	71	95	109	75	112	58	77	76	69	2	80	80	61	72	89	72	72	71	76	68	78	85	71	71
15	65	90	81	84	68	60	59	106	101	69	95	72	84	68	60	2	88	89	55	56	97	83	73	78	68	69	67	88	65	93
16	72	92	70	74	77	74	67	92	98	64	89	70	70	67	75	2	89	95	59	76	100	81	88	88	57	58	60	91	80	106
17	70	75	65	54	68	74	59	103	89	69	77	51	66	90	67	2	84	90	59	84	107	71	75	105	68	56	63	72	85	97
18	58	63	73	60	67	60	59	95	91	66	71	60	82	82	70	2	71	73	66	97	96	60	76	77	60	63	68	70	75	72
19	56	56	60	61	63	50	57	85	61	71	62	47	57	71	59	2	71	50	48	62	77	50	66	60	49	57	55	75	50	65
20	61	67	59	56	48	56	59	71	54	56	68	47	56	61	50	2	61	57	42	53	62	49	65	55	50	62	55	73	67	62
21	63	81	50	51	50	57	61	84	57	58	61	43	53	51	54	6	71	57	51	48	65	54	68	67	46	52	62	61	64	50
22	58	75	56	56	56	61	58	63	67	53	48	39	69	59	56	13	62	55	49	52	63	60	54	56	40	52	69	42	52	49
23	47	69	59	59	55	78	69	66	63	50	59	45	67	57	40	36	58	45	53	63	55	58	48	46	55	40	59	39	46	36

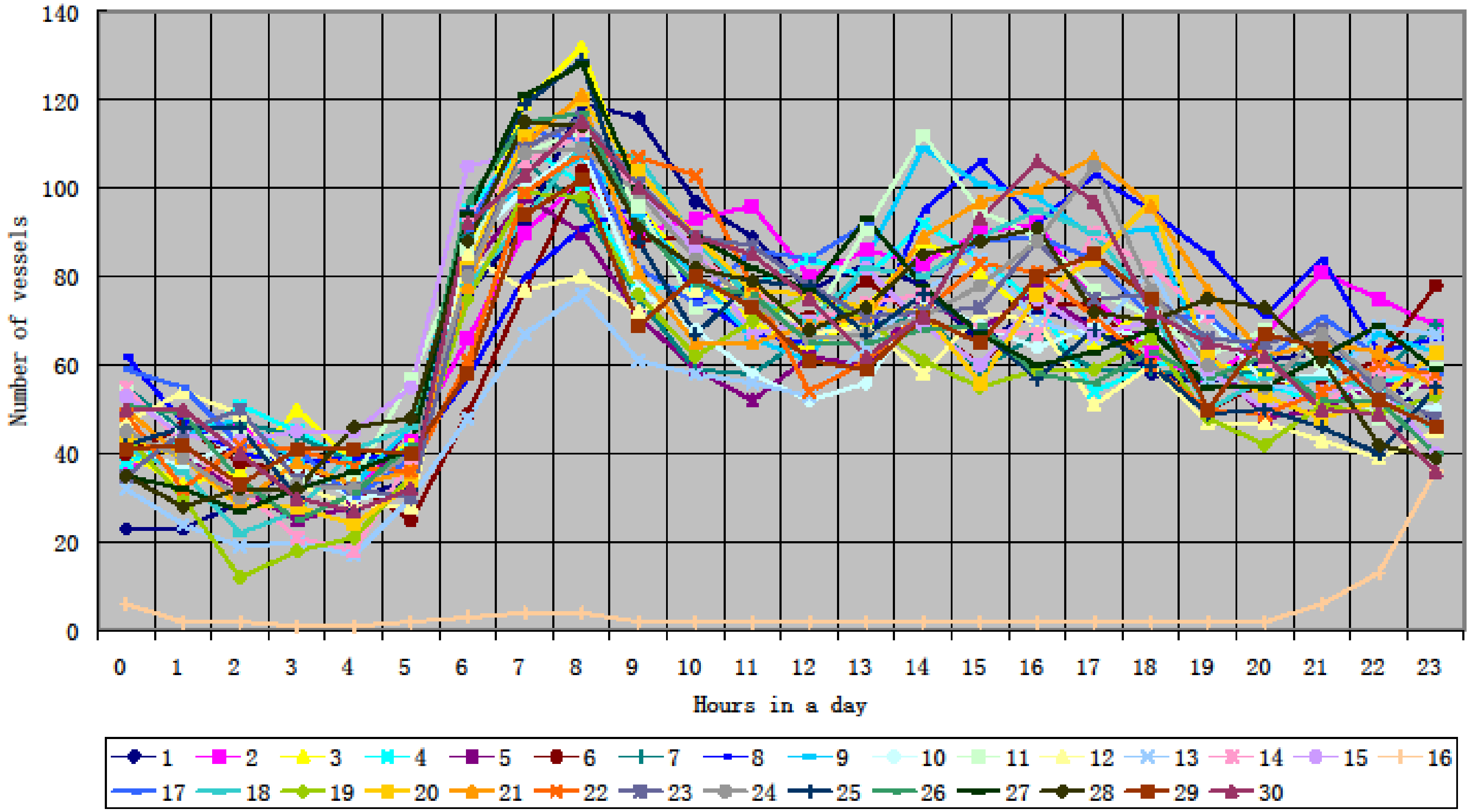


Figure C.2 Number of vessels in the seaport per hour in 30 days.

Appendix D
Core codes for simulation implementation

A C++ library has been implemented as a building ground for the seaport simulation system. This part lists the important C++ classes in the library. Some selected important codes will be listed here.

D.1 Selected C++ classes

The C++ library is an important part of this research. For convenience, all the classes developed here are named with a prefix as lq-, which is the abbreviation of this researcher's full name.

Classes for the seaport

The seaport is defined by the class named lqSeaport. Five components are included: lqLinkSet, lqNodeSet, lqShipSet, lqMap, and lqSimulator. The simulator contains a tree structure for the schedule chart: lqScheduleTree. One node on the tree is a schedule chart: lqSchedule. One schedule in the schedule chart is defined by lqShipSchedule, which contains the ship (lqShip), the route (lqRoute), and the arrival time (m_startTime).

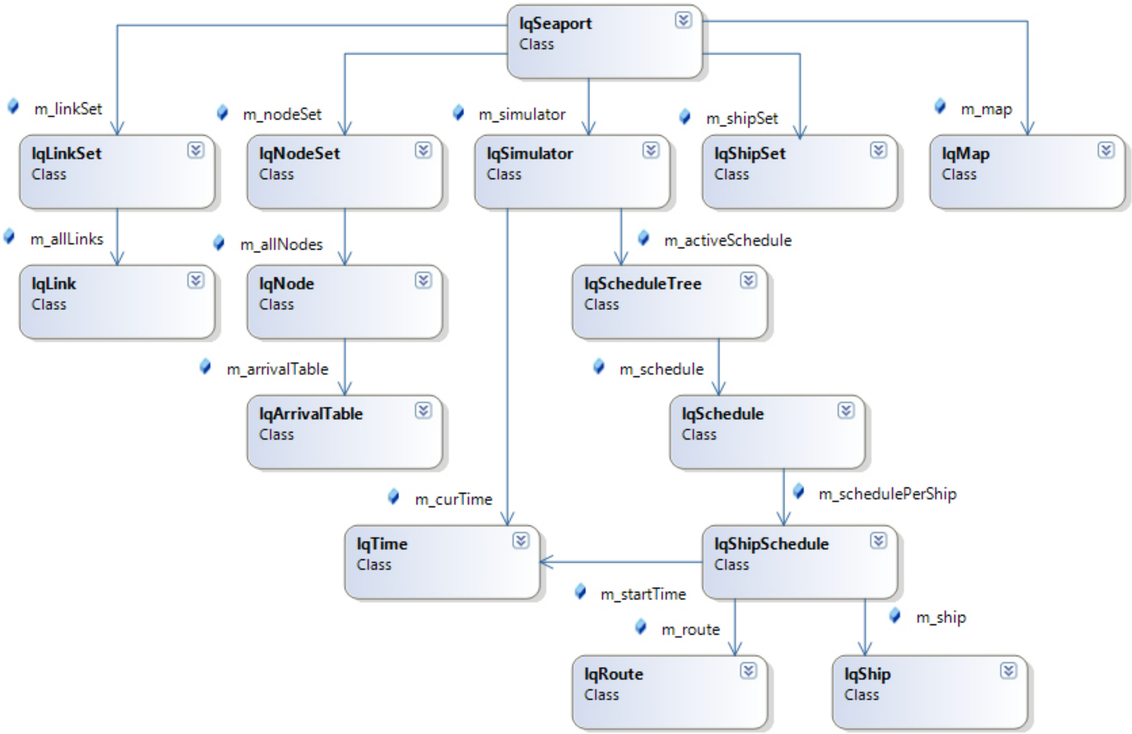
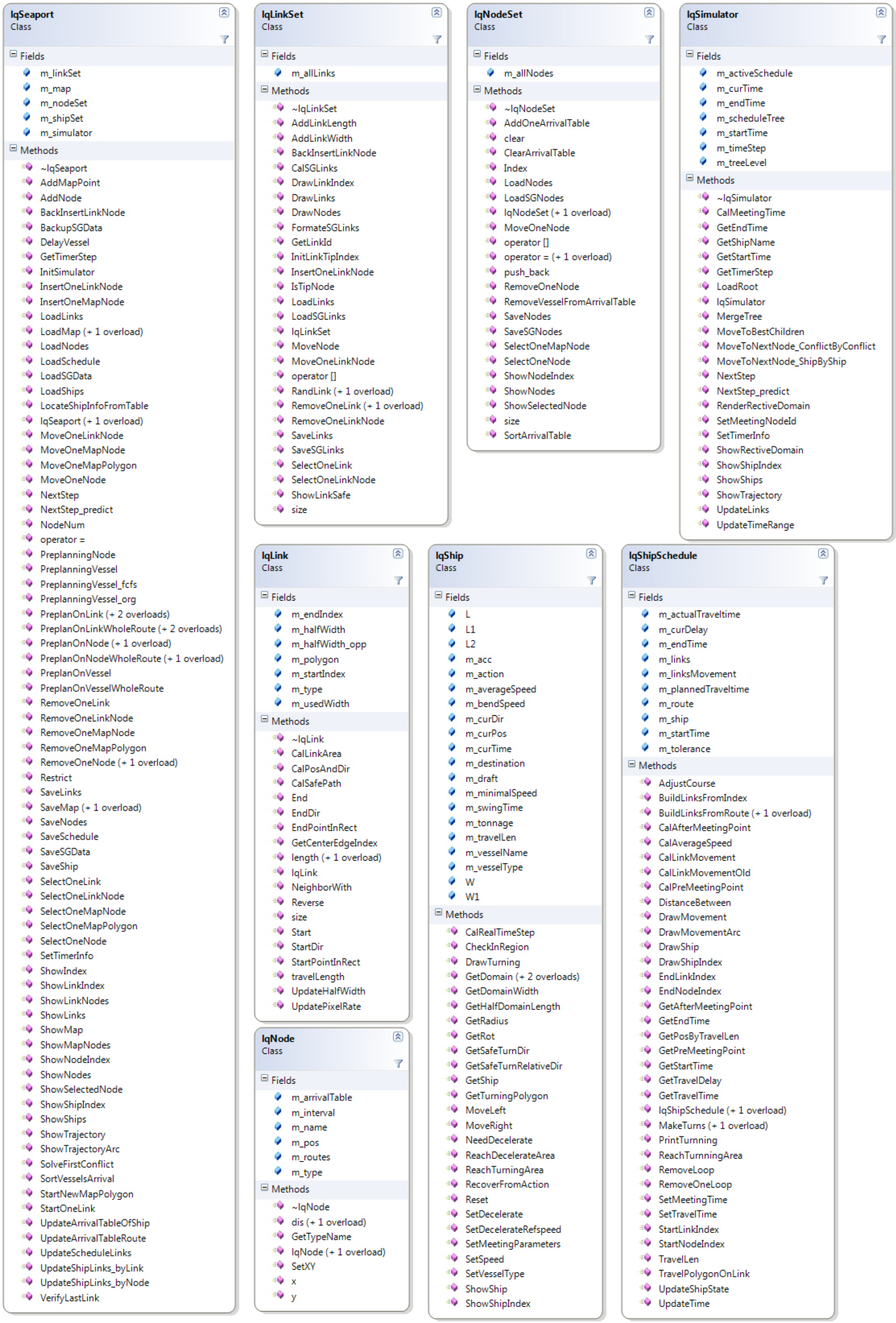


Figure D.1 Class diagram for the seaport.



IqSeaport
Class

Fields
- m_linkSet
- m_map
- m_nodeSet
- m_shipSet
- m_simulator

Methods
- ~IqSeaport
- AddMapPoint
- AddNode
- BackInsertLinkNode
- BackupSGData
- DelayVessel
- GetTimerStep
- InitSimulator
- InsertOneLinkNode
- InsertOneMapNode
- LoadLinks
- LoadMap (+ 1 overload)
- LoadNodes
- LoadSchedule
- LoadSGData
- LoadShips
- LocateShipInfoFromTable
- IqSeaport (+ 1 overload)
- MoveOneLinkNode
- MoveOneMapNode
- MoveOneMapPolygon
- MoveOneNode
- NextStep
- NextStep_predict
- NodeNum
- operator =
- PreplanningNode
- PreplanningVessel
- PreplanningVessel_fcfs
- PreplanningVessel_org
- PreplanOnLink (+ 2 overloads)
- PreplanOnLinkWholeRoute (+ 2 overloads)
- PreplanOnNode (+ 1 overload)
- PreplanOnNodeWholeRoute (+ 1 overload)
- PreplanOnVessel
- PreplanOnVesselWholeRoute
- RemoveOneLink
- RemoveOneLinkNode
- RemoveOneMapNode
- RemoveOneMapPolygon
- RemoveOneNode (+ 1 overload)
- Restrict
- SaveLinks
- SaveMap (+ 1 overload)
- SaveNodes
- SaveSchedule
- SaveSGData
- SaveShip
- SelectOneLink
- SelectOneLinkNode
- SelectOneMapNode
- SelectOneMapPolygon
- SelectOneNode
- SetTimerInfo
- ShowIndex
- ShowLinkIndex
- ShowLinkNodes
- ShowLinks
- ShowMap
- ShowMapNodes
- ShowNodeIndex
- ShowNodes
- ShowSelectedNode
- ShowShipIndex
- ShowShips
- ShowTrajectory
- ShowTrajectoryArc
- SolveFirstConflict
- SortVesselsArrival
- StartNewMapPolygon
- StartOneLink
- UpdateArrivalTableOfShip
- UpdateArrivalTableRoute
- UpdateScheduleLinks
- UpdateShipLinks_byLink
- UpdateShipLinks_byNode
- VerifyLastLink

IqLinkSet
Class

Fields
- m_allLinks

Methods
- ~IqLinkSet
- AddLinkLength
- AddLinkWidth
- BackInsertLinkNode
- CalSGLinks
- DrawLinkIndex
- DrawLinks
- DrawNodes
- FormateSGLinks
- GetLinkId
- InitLinkTipIndex
- InsertOneLinkNode
- IsTipNode
- LoadLinks
- LoadSGLinks
- IqLinkSet
- MoveNode
- MoveOneLinkNode
- operator []
- RandLink (+ 1 overload)
- RemoveOneLink (+ 1 overload)
- RemoveOneLinkNode
- SaveLinks
- SaveSGLinks
- SelectOneLink
- SelectOneLinkNode
- ShowLinkSafe
- size

IqNodeSet
Class

Fields
- m_allNodes

Methods
- ~IqNodeSet
- AddOneArrivalTable
- clear
- ClearArrivalTable
- Index
- LoadNodes
- LoadSGNodes
- IqNodeSet (+ 1 overload)
- MoveOneNode
- operator []
- operator = (+ 1 overload)
- push_back
- RemoveOneNode
- RemoveVesselFromArrivalTable
- SaveNodes
- SaveSGNodes
- SelectOneMapNode
- SelectOneNode
- ShowNodeIndex
- ShowNodes
- ShowSelectedNode
- size
- SortArrivalTable

IqSimulator
Class

Fields
- m_activeSchedule
- m_curTime
- m_endTime
- m_scheduleTree
- m_startTime
- m_timeStep
- m_treeLevel

Methods
- ~IqSimulator
- CalMeetingTime
- GetEndTime
- GetShipName
- GetStartTime
- GetTimerStep
- LoadRoot
- IqSimulator
- MergeTree
- MoveToBestChildren
- MoveToNextNode_ConflictByConflict
- MoveToNextNode_ShipByShip
- NextStep
- NextStep_predict
- RenderRectiveDomain
- SetMeetingNodeId
- SetTimerInfo
- ShowRectiveDomain
- ShowShipIndex
- ShowShips
- ShowTrajectory
- UpdateLinks
- UpdateTimeRange

IqLink
Class

Fields
- m_endIndex
- m_halfWidth
- m_halfWidth_opp
- m_polygon
- m_startIndex
- m_type
- m_usedWidth

Methods
- ~IqLink
- CalLinkArea
- CalPosAndDir
- CalSafePath
- End
- EndDir
- EndPointInRect
- GetCenterEdgeIndex
- length (+ 1 overload)
- IqLink
- NeighborWith
- Reverse
- size
- Start
- StartDir
- StartPointInRect
- travelLength
- UpdateHalfWidth
- UpdatePixelRate

IqNode
Class

Fields
- m_arrivalTable
- m_interval
- m_name
- m_pos
- m_routes
- m_type

Methods
- ~IqNode
- dis (+ 1 overload)
- GetTypeName
- IqNode (+ 1 overload)
- SetXY
- x
- y

IqShip
Class

Fields
- L
- L1
- L2
- m_acc
- m_action
- m_averageSpeed
- m_bendSpeed
- m_curDir
- m_curPos
- m_curTime
- m_destination
- m_draft
- m_minimalSpeed
- m_swingTime
- m_tonnage
- m_travelLen
- m_vesselName
- m_vesselType
- W
- W1

Methods
- CalRealTimeStep
- CheckInRegion
- DrawTurning
- GetDomain (+ 2 overloads)
- GetDomainWidth
- GetHalfDomainLength
- GetRadius
- GetRot
- GetSafeTurnDir
- GetSafeTurnRelativeDir
- GetShip
- GetTurningPolygon
- MoveLeft
- MoveRight
- NeedDecelerate
- ReachDecelerateArea
- ReachTurningArea
- RecoverFromAction
- Reset
- SetDecelerate
- SetDecelerateRefspeed
- SetMeetingParameters
- SetSpeed
- SetVesselType
- ShowShip
- ShowShipIndex

IqShipSchedule
Class

Fields
- m_actualTraveltime
- m_curDelay
- m_endTime
- m_links
- m_linksMovement
- m_plannedTraveltime
- m_route
- m_ship
- m_startTime
- m_tolerance

Methods
- AdjustCourse
- BuildLinksFromIndex
- BuildLinksFromRoute (+ 1 overload)
- CalAfterMeetingPoint
- CalAverageSpeed
- CalLinkMovement
- CalLinkMovementOld
- CalPreMeetingPoint
- DistanceBetween
- DrawMovement
- DrawMovementArc
- DrawShip
- DrawShipIndex
- EndLinkIndex
- EndNodeIndex
- GetAfterMeetingPoint
- GetEndTime
- GetPosByTravelLen
- GetPreMeetingPoint
- GetStartTime
- GetTravelDelay
- GetTravelTime
- IqShipSchedule (+ 1 overload)
- MakeTurns (+ 1 overload)
- PrintTurnning
- ReachTurnningArea
- RemoveLoop
- RemoveOneLoop
- SetMeetingTime
- SetTravelTime
- StartLinkIndex
- StartNodeIndex
- TravelLen
- TravelPolygonOnLink
- UpdateShipState
- UpdateTime

Figure D.2 Selected seaport related classes.

Classes for data processing

Data processing need five types of processing: routes (lqFromToRouteSet, lqFromToRoute), names per node (lqNodeNameSet, lqNodeName), schedules of one month (lqSGResourceSet, lqSgResource, lqSGResourceItem), vessel’s speed types (lqShipSpeedTypeSet, lqShipSpeedType), and vessel’s types (lqShipTypeSet, lqShipType).

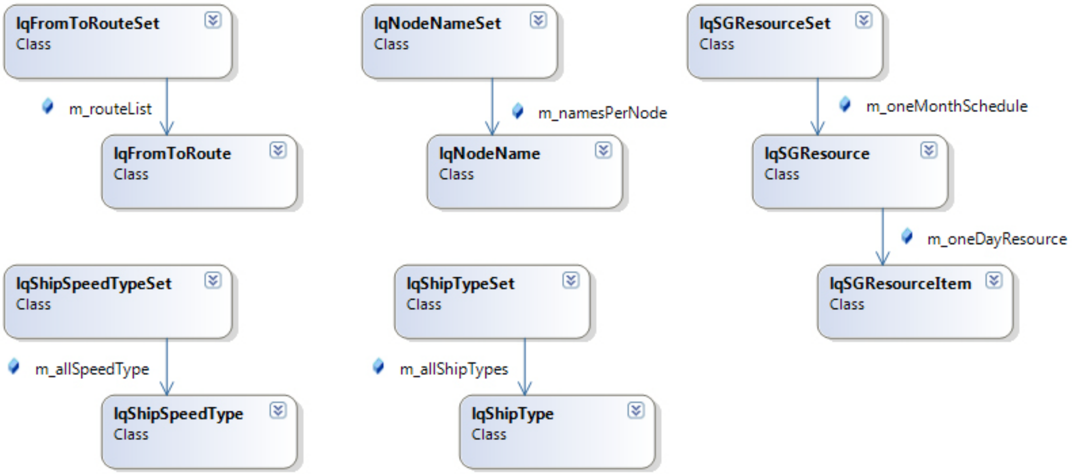


Figure D.3 Class diagram for data processing.

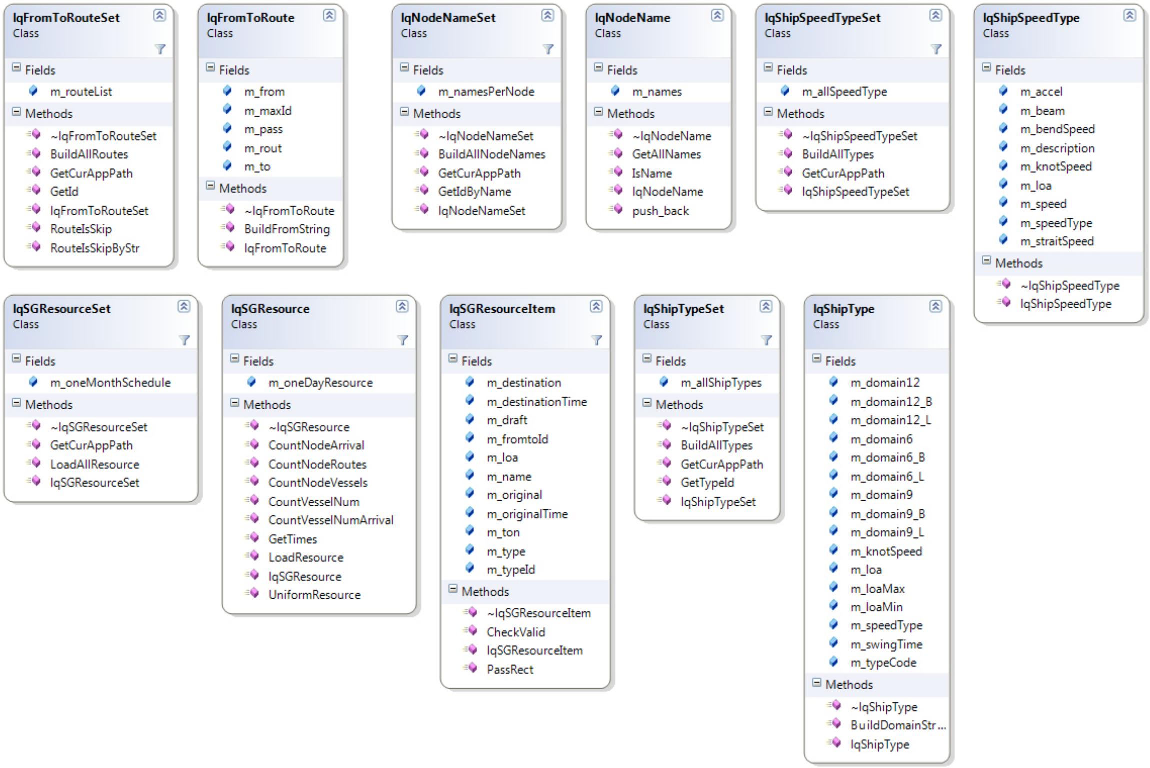


Figure D.4 Selected classes for data processing.

Classes for conflict detection

This part includes a conflict detector (lqConflictDector), a conflict condition (lqConflictCondiction), the conflict notation (lqConflict), and the corresponding delay (lqDelay).

Figure D.5 Class diagram for conflict detection.

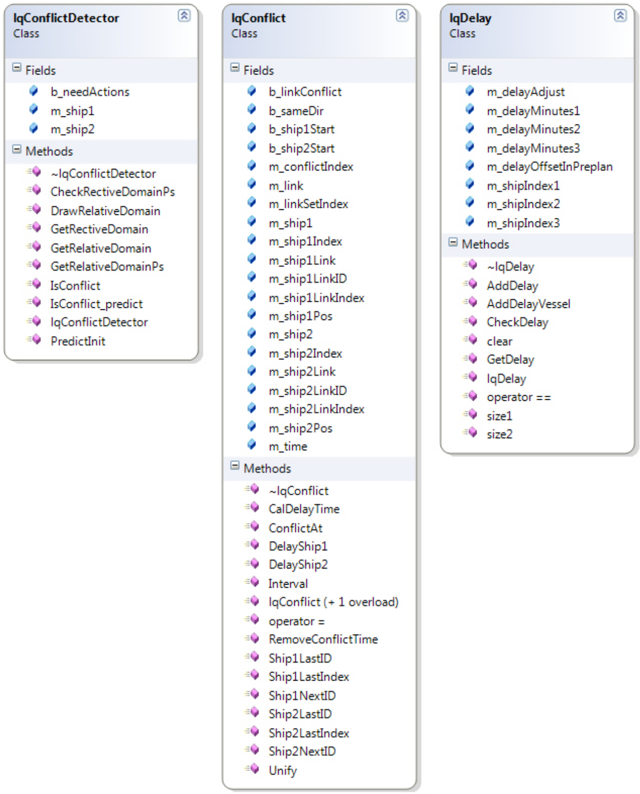


Figure D.6 Selected classes for conflict detection.

Classes for conflict resolution

During the simulation, a vessel can takes actions to avoid conflicts. This is implemented in lqShipAction, including making a turn (lqTurning), deceleration (lqShipDeceleration), and overtaking (lqShipParallel).

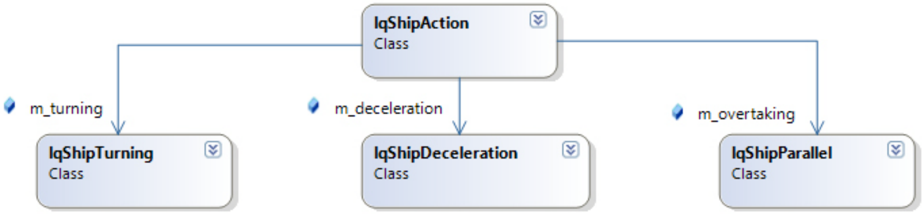


Figure D.7 Class diagram for conflict resolution.

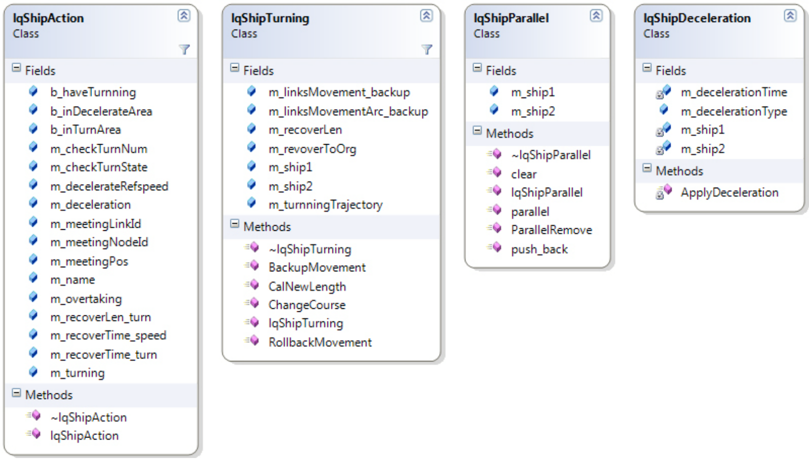


Figure D.8 Selected classes for conflict resolution.

Classes for mathematical calculation

Some mathematical calculations are basic for conflict detection/prediction and conflict resolution, including distance calculation (lqDistance), in-out testing (lqInOutTest), intersection checking (lqIntersection), and projection calculation (lqProjection).

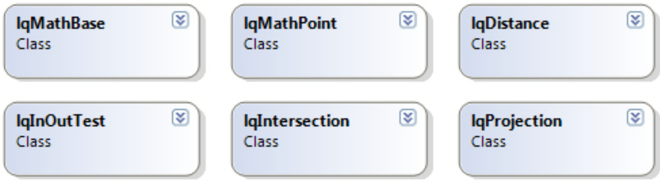


Figure D.9 Class diagram for mathematical calculation.

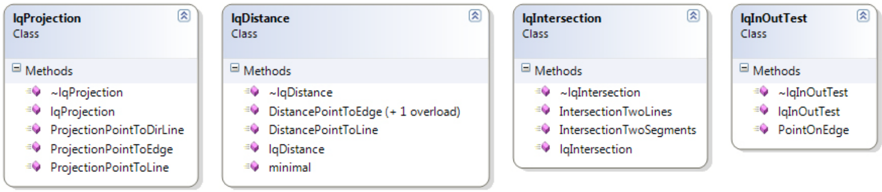


Figure D.10 Selected classes for mathematical calculation.

Classes for trajectory calculation

Vessel's trajectory is calculated using lqShipTrajectory. Each trajectory is represented as a polygon (lqPolygon). Each polygon is composed of points (lqPoint).

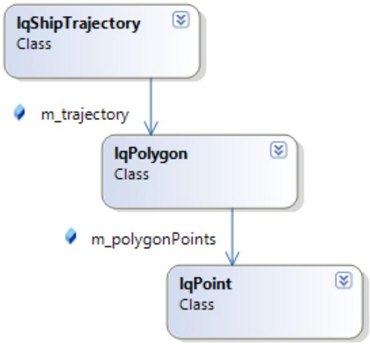


Figure D.11 Class diagram for trajectory calculation.

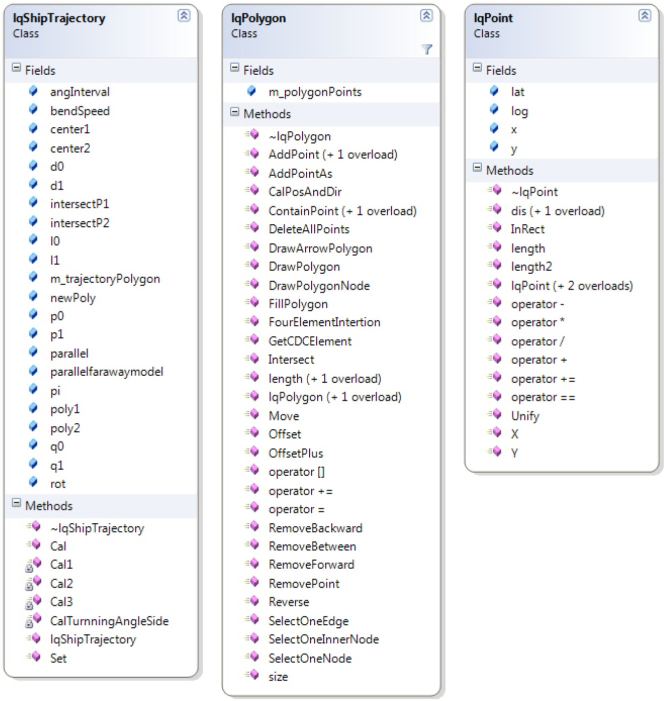


Figure D.12 Selected classes for trajectory calculation.

Classes for framework

There are some other functions needed in the simulation system, such as rendering (lqCDC), capturing screens (lqBitmap), and file opening & saving (lqFiles).

Figure D.13 Class diagram for framework.

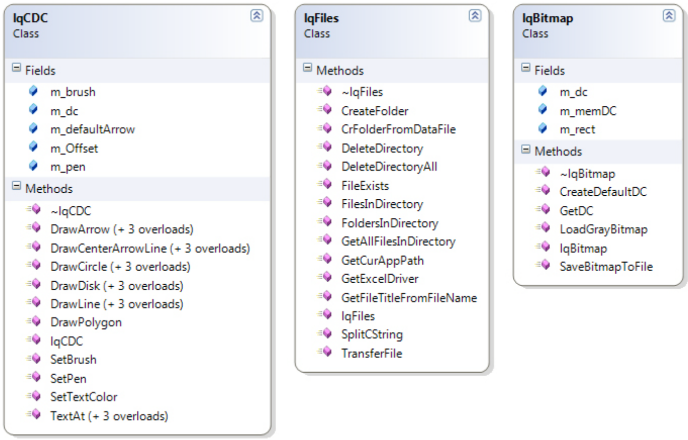


Figure D.14 Selected classes for framework.

D.2 Selected C++ codes

Source codes for conflict detection/prediction

```
// get the relative movements between two ships
void lqConflictDetector::GetRelativeDomainPs( lqShip& ship2,
                                              lqShip& ship1,
                                              vector<lqPolygon>& rdomain,
                                              vector<lqPolygon>& refdomain)
{
   rdomain.clear();
   refdomain.clear();

   // ship1 first, then ship2
   // cal the relative domain of ss1 with respective to ship2
   lqShip ss1,ss2;
   if (ship1.m_meetingTime < ship2.m_meetingTime)
   {
      ss1 = ship1;
      ss2 = ship2;
   }
   else
   {
      ss1 = ship2;
      ss2 = ship1;
   }

   vector<double> t;
   vector<lqMathPoint> v1;
   vector<lqMathPoint> v2;
   vector<double>& t1 = ss1.psTime;
   vector<double>& t2 = ss2.psTime;
   vector<lqMathPoint>& v11 = ss1.psVelocity;
   vector<lqMathPoint>& v12 = ss2.psVelocity;
   int t1id = 0;
   int t2id = 0;
   int v1id = 0;
   int v2id = 0;

   size_t si = t1.size()+t2.size();
   t.reserve(si);
   v1.reserve(si);
   v2.reserve(si);
   rdomain.reserve(si*4);
   refdomain.reserve(si*4);

   // save starting
   t.push_back(0);
   v1.push_back(v11[0]);
   v2.push_back(v12[0]);
```

```
    while (t1id < t1.size() || t2id < t2.size())
    {
        if (t1id == t1.size()||t2id == t2.size())
        {
            break;
        }

        if (t1id >= t1.size()
            || t2id >= t2.size())
        {
            break;
        }

        if (t2[t2id]>t1[t1id])
        {
            v1id++; // ss1 turns at this point
            if (v1id >= v11.size())
            {
                v1id--;
            }
            t.push_back(t1[t1id]);
            v1.push_back(v11[v1id]);
            v2.push_back(v12[v2id]);
            t1id++;
        }
        else
        {
            v2id++; // ss2 turns at this point
            if (v2id >= v12.size())
            {
                v2id--;
            }
            t.push_back(t2[t2id]);
            v1.push_back(v11[v1id]);
            v2.push_back(v12[v2id]);
            t2id++;
        }
    }

    t.pop_back();
    lqPolygon poly,polyc,rp;
    for (size_t l = 1; l < t.size(); l++)
    {
        double t0 = t[l-1];
        double t1 = t[l];
        lqMathPoint w = v1[l-1]-v2[l-1];
        double c = t1-t0;
        if (c<=0 || (w.x == 0 && w.y==0) )
        {
            continue;
        }

        if (l>=2)
        {
            ss1.m_curDir = v1[l-1].GetPoint();
            ss2.m_curDir = v2[l-1].GetPoint();
        }

        poly = ss1.GetDomain();
        polyc = ss2.GetDomain();

        for (int iii = 0 ; iii < 4; iii++)
        {
            rdomain.push_back(GetRelativeDomain(poly,iii,w,c));
            refdomain.push_back(polyc);
        }

        ss1.m_curPos = ss1.m_curPos + (w*c/lqSeaport::m_curPixelSize).GetPoint();
    }
}

// predict the conflict between two ships
```

```
BOOL lqConflictDetector::CheckRectiveDomainPs(lqShip& s1,lqShip& s2)
{
    vector<lqPolygon> rdomain,refdomain;
    GetRelativeDomainPs(s1,s2,rdomain,refdomain);

    lqPolygon rp,polyc;
    for (size_t iii = 0 ; iii < rdomain.size(); iii++)
    {
        rp = rdomain[iii];
        polyc = refdomain[iii];
        if (lqPolygon::Intersect(polyc,rp))
        {
            rdomain.clear();
            refdomain.clear();
            return TRUE;
        }
    }
    rdomain.clear();
    refdomain.clear();
    return FALSE;
}


// detect whether the two ships conflict
BOOL lqConflictDetector::IsConflict()
{
    if (m_ship1.b_finish || m_ship2.b_finish
        || !m_ship1.b_visable || !m_ship2.b_visable)
    {
        return FALSE;
    }

    lqPolygon poly1 = m_ship1.GetDomain();
   lqPolygon poly2 = m_ship2.GetDomain();

    // check whether the two domains intersect
    return lqPolygon::Intersect(poly1,poly2);
}

// predict whether the two ships conflict
BOOL lqConflictDetector::IsConflict_predict()
{
    if (m_ship1.b_finish || m_ship2.b_finish
        || !m_ship1.b_visable || !m_ship2.b_visable)
    {
        return FALSE;
    }

     return CheckRectiveDomainPs(m_ship1,m_ship2);
}
```

Source codes for preplanning

```
// vessel-based preplanning, with first-come-first-serve
void lqSeaport::PreplanningVessel_fcfs()
{
    vector<lqShipSchedule>& shipSchedule = m_simulator.m_scheduleTree.m_schedule.m_shipSchedule;
    size_t num = shipSchedule.size();
    for (size_t i = 0; i < num; i++)
    {
        UpdateArrivalTableOfShip(i);
        shipSchedule[i].m_curDelay = 0;
    }

    // sort arrival table
    m_nodeSet.SortArrivalTable();
    size_t timeadj = 0;

    while (1)
    {
        BOOL vesselAdjusted = FALSE;
```

```
        for (size_t kl = 0; kl < shipSchedule.size(); kl++)
        {
            if (b_toleranceInPreplan)
            {
                int d = shipSchedule[kl].m_tolerance-shipSchedule[kl].m_curDelay;
                if (lqSysConfig::m_preplanType == PREPLAN_TOR)
                {
                    // if accept delay tolerance in preplaning
                    d = (int)(shipSchedule[kl].m_tolerance -shipSchedule[kl].m_curDelay);
                }
                if (d<=0)
                {
                    // if reach delay tolerance, skip this ship
                    continue;
                }
            }

            if (PreplanOnVesselWholeRoute((int)kl))
            {
                vesselAdjusted = TRUE;
                break;
            }
        }

        if (vesselAdjusted)
        {
            // at least one vessel is delay, continue
            continue;
        }
        // no vessel is delay. Finish
        break;
    }
}


// perform preplanning on a vessel
BOOL  lqSeaport::PreplanOnVesselWholeRoute(int vesselid)
{
    lqShipSchedule shipSchedule = m_simulator.m_scheduleTree.m_schedule.m_shipSchedule[vesselid];
    if (b_toleranceInPreplan)
    {
        // check ship's tolerance in preplanning
        int d = shipSchedule.m_tolerance-shipSchedule.m_curDelay;
        if (lqSysConfig::m_preplanType == PREPLAN_TOR)
        {
            d = (int)(shipSchedule.m_tolerance -shipSchedule.m_curDelay);

        }
        if (d<=0)
        {
            return FALSE;
        }
    }

    BOOL vesseladjust = TRUE;
    while (vesseladjust)
    {
        vesseladjust = FALSE;
        for (size_t i = 0; i < shipSchedule.m_links.size(); i++)
        {
            //check each link to guarantee the time separation
            if (PreplanOnLinkWholeRoute((int)shipSchedule.m_linkIndex[i],vesselid))
            {
                vesseladjust = TRUE;
            }
        }
        //check each node to guarantee the time separation
        if (PreplanOnNodeWholeRoute(shipSchedule.m_links[0].m_startIndex,vesselid))
        {
            vesseladjust = TRUE;
        }
        for ( i = 0; i < shipSchedule.m_links.size(); i++)
```

```
        {
            //check each node to guarantee the time separation
            if (PreplanOnNodeWholeRoute(shipSchedule.m_links[i].m_endIndex,vesselid))
            {
                vesseladjust = TRUE;
                break;
            }
        }
    }
    return vesseladjust;
}
```

Source codes for conflict resolution

```
// calculate a new trajectory for ship1 to avoid the conflict with ship2
void lqShipTurning::ChangeCourse()
{
    lqShip& ship1 = m_ship1->m_ship;
    lqShip& ship2 = m_ship2->m_ship;
    lqShipAction& action1 = ship1.m_action;
    lqShipAction& action2 = ship2.m_action;
    lqMathPoint p0,dir0,p1,dir1,q,dir,p0arc,dir0arc,p1arc,dir1arc;
    int linkid0,linkid1,edgeid0,edgeid1,linkid,edgeid,linkid0arc,linkid1arc,edgeid0arc,edgeid1arc;
    double lenp0,lenp1,lenq,lenp0arc,lenp1arc;

    double ship1len = ship1.GetHalfDomainLength()*2;
    double ship2len = ship2.GetHalfDomainLength()*2;

    double t1 = ship1len*lqSysConfig::m_turnrate;
    double t2 = t1;


    m_ship1->CalPreMeetingPoint(t1,p0,dir0,linkid0,edgeid0,lenp0);
    m_ship1->CalAfterMeetingPoint(t2,p1,dir1,linkid1,edgeid1,lenp1);
    m_ship2->CalPreMeetingPoint(ship2len,q,dir,linkid,edgeid,lenq);

    double nt1 = (action1.m_meetingPos-p0).length();
    double nt2 = (action1.m_meetingPos-p1).length();
    double nlen1 = (action1.m_meetingPos-q).length();

    double d1 = (p0-q).length();
    CalNewLength(nt1,nlen1,d1);
    double d2 = (p1-q).length();
    CalNewLength(nt2,nlen1,d2);
    nt1 *= lqSeaport::m_curPixelSize;
    nt2 *= lqSeaport::m_curPixelSize;


    if (nt1>t1)
    {
        t1 = nt1;
    }
    if (nt2>t2)
    {
        t2 = nt2;
    }
    m_ship1->CalPreMeetingPoint(t1,p0,dir0,linkid0,edgeid0,lenp0);
    m_ship1->CalAfterMeetingPoint(t2,p1,dir1,linkid1,edgeid1,lenp1);

    vector<lqLink> backup = m_ship1->m_linksMovement;
    m_ship1->m_linksMovement = m_ship1->m_linksMovementArc;
    m_ship1->CalPreMeetingPoint(t1,p0arc,dir0arc,linkid0arc,edgeid0arc,lenp0arc);
    m_ship1->CalAfterMeetingPoint(t2,p1arc,dir1arc,linkid1arc,edgeid1arc,lenp1arc);
    m_ship1->m_linksMovement = backup;

    if (linkid1-linkid0>1)
    {
        // there is a very small link
        planningError = 1;
        return;
    }
```

```
    int did = linkid1-linkid0;
    if (did==0)
    {
        m_turnningTrajectory = m_ship1->m_linksMovement[linkid0].m_polygon.RemoveBetween(p0,edgeid0,p1,edgeid1,q);
    }
    if (did==1)
    {
        m_ship1->m_linksMovement[linkid0].m_polygon.RemoveForward(p0,edgeid0);
        m_ship1->m_linksMovement[linkid1].m_polygon.RemoveBackward(p1,edgeid1);

        m_turnningTrajectory.m_points.clear();
        m_turnningTrajectory.AddPoint(p0.GetPoint());
        m_turnningTrajectory.AddPoint(q.GetPoint());
        m_turnningTrajectory.AddPoint(p1.GetPoint());
        lqMathPoint pre = p0;
        lqMathPoint after = p1;
        size_t presize = m_ship1->m_linksMovement[linkid0].m_polygon.size();
        if (presize>1)
        {
            pre = m_ship1->m_linksMovement[linkid0].m_polygon.m_points[presize-2];
        }
        size_t aftersize = m_ship1->m_linksMovement[linkid1].m_polygon.size();
        if (aftersize>1)
        {
            after = m_ship1->m_linksMovement[linkid1].m_polygon.m_points[1];
        }
        for (int ss = 0; ss < lqSysConfig::m_turnSmoothingNum; ss++)
        {
            m_turnningTrajectory.FourElementIntertion(pre,after);
        }

        for (size_t k = 1; k < m_turnningTrajectory.m_points.size(); k++)
        {
            m_ship1->m_linksMovement[linkid0].m_polygon.m_points.push_back(m_turnningTrajectory.m_points[k]);
        }

    }

    did = linkid1arc-linkid0arc;
    if (did==0)
    {
        m_ship1->m_linksMovementArc[linkid0arc].m_polygon.RemoveBetween(p0arc,edgeid0arc,p1arc,edgeid1arc,q);
    }
    if (did==1)
    {
        m_ship1->m_linksMovementArc[linkid0arc].m_polygon.RemoveForward(p0arc,edgeid0arc);
        m_ship1->m_linksMovementArc[linkid1arc].m_polygon.RemoveBackward(p1arc,edgeid1arc);
        for (size_t k = 1; k < m_turnningTrajectory.m_points.size(); k++)
        {
            m_ship1->m_linksMovementArc[linkid0arc].m_polygon.m_points.push_back(m_turnningTrajectory.m_points[k]);
        }
    }

    m_recoverLen = m_ship1->TravelLenToCorner(p1);
    double dist = m_recoverLen - m_ship1->m_ship.m_travelLen;
    double distTime = dist * lqSeaport::m_curPixelSize / m_ship1->m_ship.m_averageSpeedDefault;
    m_revoverToOrg = m_ship1->m_ship.m_curTime + CTimeSpan((int)distTime);
}


 // i-th ship makes a turn to avoid the conflict with j-th ship
// check whether this i-th ship can make this turn
// if i-th ship conflicts with other ships in the turnning course, i-th ship can not make the turn. It will decelerates.
BOOL lqScheduleTree::ValidTurn(lqTime time, int i , int j)
{
    vector<lqShipSchedule>& ships = m_schedule.m_shipSchedule;
    lqShipSchedule& schedule1 = ships[i];

    // backup conflicts
    vector<lqConflict> conflicts_predict = m_conflicts_predict;
    m_conflicts_predict.clear();
```

```
    // check new conflicts
    BOOL validturn = TRUE;
    for (size_t k = 0; k < m_schedule.m_shipSchedule.size(); k++)
    {
        if (k == i || k == j)
        {
            continue;
        }

        if(DetectConflicts_predict(time,i,k,FALSE))
        {
            DetectConflicts_predict(time,k,i,FALSE);
        }

        for (size_t l = 0; l < m_conflicts_predict.size(); l++)
        {
            if (!m_conflicts_predict[l].b_linkConflict
                && m_conflicts_predict[l].m_conflictIndex == schedule1.m_ship.m_action.m_meetingNodeId)
            {
                // set to orginal
                m_conflicts_predict = conflicts_predict;

                return FALSE;
            }
        }
    }
    // set to orginal
    m_conflicts_predict = conflicts_predict;

    return TRUE;
}
```