# MULTI-OBJECTIVE ROUTE SELECTION FOR ICE-CLASS VESSELS USING REINFORCEMENT LEARNING AND GRAPH-BASED APPROACHES

By © Trung Tien Tran

A thesis submitted to the School of Graduate Studies in partial fulfillment of the requirements for the degree of

**Master of Engineering**

**Computer Engineering**

**Engineering & Applied Science**

Memorial University of Newfoundland

**October 2021**

St. John's, Newfoundland & Labrador

# Abstract

Route selection for ships in ice is a complicated problem in marine navigation. The navigators have to optimize many economic and environmental factors of the routes while adhering to all maritime regulations to ensure safety. The International Maritime Organization has introduced the Polar Operational Limit Assessment Risk Indexing System (POLARIS) as guidelines for all vessels operating in the Arctic Ocean. This research investigates a framework for finding an optimal route for different ice-class vessels using two methods: graph-based approaches and reinforcement learning. The system uses ice charts from the Canadian Ice Service to explore possible routes in a grid world. Reward and cost functions are formulated to achieve operational objectives, such as optimizing the distance travelled, voyage time, and fuel consumption while complying with POLARIS regulation. The graph-based method surpasses the Q-learning in deterministic cases. Despite the shortcoming of not handling the non-deterministic environment, it also shows similar routes compared to Q-learning in a stochastic context. The trial results show that the framework provides a means to identify an optimal route for vessels navigating through ice-covered waters.

# Acknowledgements

Firstly, the author would like to express gratitude to the financial support of the National Sciences and Engineering Research Council (NSERC) - Husky Energy Industrial Research Chair in Safety at Sea.

The author also likes to extend the most profound appreciation to his co-supervisors: Dr. Brian Veitch, Dr. Dennis Peters, and Dr. Mashrura Musharraf. This thesis would not have been completed without their continuous advice and feedback.

The author acknowledges the great support of Thomas Browne. His domain knowledge contributes much to the research. He also helps conduct the validation section with two experienced captains. The author also acknowledges other colleagues in the Safety at Sea group, especially Jennifer Smith, Vahid Salehi, and Jonathan Soper, for creating a great place to work.

Last but not least, the author gives special thankfulness to his family and friends, especially Phuong Tran, Sinh Bui, Delphina and Paul Mercer, for warming his heart in the bitterly cold weather of Newfoundland.

# Table of Contents

**Abstract** ....................................................................................................................ii

**Acknowledgements** ....................................................................................................... iii

**Table of Contents**.........................................................................................................iv

**List of Figures** ...........................................................................................................vi

**List of Tables**..............................................................................................................ix

**Lists of Abbreviations** ................................................................................................xi

**Chapter 1: Introduction**...................................................................................................1

**Chapter 2: Literature Review**............................................................................................3

1. **Overview**.....................................................................................................................3
2. **Objectives** ...................................................................................................................3
3. **Constraints** ..................................................................................................................4
4. **Optimization method**......................................................................................................5
5. **Summary** ....................................................................................................................10

**Chapter 3: Framework and Methods** .................................................................................11

1. **Framework**..................................................................................................................11
2. **Reinforcement learning**................................................................................................25
3. **Graph-based method**.....................................................................................................32

**Chapter 4: Trials and Results** .......... 36

1. **Deterministic environment** .......... 36
2. **Non-deterministic environment** .......... 41
3. **Route from Goose Bay to St. John's** .......... 45
4. **Route from Goose Bay to St. John's with temporal change of the environment** .......... 53
5. **Validation** .......... 63

**Chapter 5: Discussion** .......... 67

1. **Optimality** .......... 67
2. **Reinforcement learning and graph-based methods** .......... 69
3. **Deterministic and non-deterministic environment** .......... 70
4. **Temporal dynamics of the environment** .......... 71
5. **Validation with two captains** .......... 71

**Chapter 6: Conclusion and Future Work** .......... 74

**References** .......... 76

# List of Figures

Figure 1. Overall framework (Tran et al., 2020)............................................................ 11

Figure 2. An example of an ice regime.............................................................................. 14

Figure 3. An ice chart of Newfoundland waters on March 11, 2020 (Ice Archive, 2019) 15

Figure 4. The discretized ice chart of Newfoundland waters on March 11, 2020............ 16

Figure 5. Reinforcement learning model (Tran et al., 2020) ........................................... 25

Figure 6. The connectivity at a node in the basic graph .................................................. 32

Figure 7. The connectivity at a node in multi-graph........................................................ 33

Figure 8. A simulated ice chart ......................................................................................... 37

Figure 9. The route suggested for PC5 (left: Dijkstra's algorithm, right: RL) ................. 39

Figure 10. The route suggested for PC7 (left: Dijkstra's algorithm, right: RL) ............... 39

Figure 11. The route suggested for IC (left: Dijkstra's algorithm, right: RL) .................. 39

Figure 12. An example of ice drifting................................................................................ 42

Figure 13. Example of environment when p = 0 (left), and p = 1 (right) ......................... 42

Figure 14. The route suggested for PC5 (left: Dijkstra's algorithm, right: RL) ............... 43

Figure 15. The route suggested for PC7 (left: Dijkstra's algorithm, right: RL) ............... 43

Figure 16. The route suggested for IC (left: Dijkstra's algorithm, right: RL) ................. 43

Figure 17. Route of PC5 in the deterministic environment ............................................. 49

Figure 18. Route of PC7 in a deterministic environment ................................................ 50

Figure 19. Route of PC5 with ice drifting......................................................................... 51

Figure 20. Route of PC7 with ice drifting......................................................................... 52

Figure 21. The ice chart of Newfoundland waters on Mar 12, 2020 (Ice Archive, 2019) 55

Figure 22. The ice chart of Newfoundland waters on Mar 13, 2020 (Ice Archive, 2019) 56

Figure 23. Route for March 11, 2020 – day 1................................................................... 57

Figure 24. Route for March 12, 2020 – day 2................................................................... 58

Figure 25. Route for March 13, 2020 – day 3................................................................... 59

Figure 26. Full 3-day trip from Goose Bay to St. John's.................................................. 60

Figure 27. Comparison to old route (green: common area, red: new route, grey: old route) ........................................................................................................................................ 61

Figure 28. Ice chart of Newfoundland on March 11, 2020 and route selected by Expert 1 ........................................................................................................................................ 65

Figure 29. Ice chart of Newfoundland on March 11, 2020 and route selected by Expert 2 ........................................................................................................................ 66

# List of Tables


Table 1. Coding of ice types (Canadian Ice Service, 2016)............................................ 13

Table 2. An example of fuel consumption on an ice regime ........................................... 19

Table 3. Risk index values of different ice types and ice classes (IMO, 2016)................ 21

Table 4. A proposed RIV table is used for the CIS ice chart........................................... 22

Table 5. Operational criteria imposed by POLARIS (IMO, 2016).................................. 23

Table 6. Safe speed limits for elevated risk operations (IMO, 2016)............................... 24

Table 7. List of actions in the reinforcement learning model .......................................... 27

Table 8. The ice concentration of all ice types of the simulated ice chart (per tenth) ...... 38

Table 9. The RIO of all ice regime of the simulated ice chart......................................... 38

Table 10. Running time (seconds) ..................................................................... 40

Table 11. Running time of the environment with ice drifting (seconds) .......................... 44

Table 12. The ice concentration of all ice types of the Newfoundland ice chart.............. 46

Table 13. RIO calculation of ice regimes of the ice chart on March 11, 2020 ................. 48

Table 14. Metrics of different routes for PC5 and PC7 ................................................... 48

Table 15. New RIO on day 2 .......... 54

Table 16. New RIO on day 3 .......... 54

Table 17. Ship particulars (Frederking, 2003) .......... 63

# Lists of Abbreviations

| | |
|---|---|
| AIRSS | Arctic Ice Regime Shipping System |
| ASPPR | Arctic Shipping Pollution Prevention Regulations |
| CISIMO | Canadian Ice Service |
| IMO | International Maritime Organization |
| POLARIS | Polar Operational Limit Assessment Risk Indexing System |
| RIO | Risk Index Outcome |
| RIV | Risk Index Value |
| RL | Reinforcement Learning |

# Chapter 1: Introduction

Pathfinding is an essential step of shipping operations in ice-covered water. The ultimate goal of this activity is to find a route that optimizes operational objectives. These objectives usually are distance travelled, voyage time, and fuel consumption. Besides optimizing the operational goals, the ships need to comply with all safety regulations, such as Polar Operational Limit Assessment Risk Indexing System (POLARIS) (International Maritime Organization [IMO], 2016) and Arctic Ice Regime Shipping System (AIRSS) (Transport Canada, 2017).

The topic has attracted the attention of several researchers for years. Early work on this problem was reported by Frederking (2003). In that research, two potential routes were compared to beat each other for a voyage in the Gulf of St. Lawrence, Canada. The objectives were to optimize time and fuel consumption. Though the method is simple, the work required that some route candidates be selected in advance for manual comparison. In reality, multiple possible routes connect two points, which leads to the fact that this approach is not efficient. Other works use optimization-based techniques. They establish a cost function of the route and use Powell’s method (Kotovirta et al., 2009) and the finite element method (Piehl et al., 2017) to find the best ways by minimizing the cost function. Graph search is an effective method to solve the pathfinding problem, which is used by Guiness et al. (2014), Choi et al. (2015), Liu et al. (2016), and Lehtola et al. (2019) to navigate the ship in ice. The works mentioned above can find the optimal routes for a ship, but the gaps still exist. While Frederking (2003), Kotovirta et al. (2009), Piehl et al. (2017), Guiness et al. (2014), Choi et al. (2015) do not take safety restrictions induced by ice into consideration, Liu et al. (2016) do not include the speed along the route. The most recent research of Lehtola et al. (2019) addresses these problems by setting preference rules so that the routes are always safe and indicate a speed map for the ship, but it ignores the fuel consumption in the model.

The present research solves the pathfinding problem for ice-class ship pathfinding using two methods: graph search and reinforcement learning. This research aims to investigate a general framework for route selection where multiple conflicting objectives, such as distance, time, and fuel consumption, are optimized. At the same time, the operation strictly adheres to the safety regulations, namely POLARIS, in the Arctic Ocean. The current work on route selection for vessels in ice under POLARIS constraint is novel. Early results of this work are presented in (Tran et al., 2020).

The layout of the thesis is organized into six chapters. Chapter 1 introduces the problem. Chapter 2 reviews the related works of pathfinding for ships in ice with a multi-objective context. Chapter 3 presents the conceptual framework to explore the optimal route for ice-class vessels. Chapter 4 demonstrates the method by applying it to realistic scenarios and comparing the results with expert navigators. Chapter 5 covers the discussion. Chapter 6 is the conclusion with future work.

# Chapter 2: Literature Review

### 1. Overview

Pathfinding plays a vital role in navigation for ice-class vessels in the ocean. This work ensures the selected route is optimized for multiple competing objectives, such as minimizing the distance travelled, minimizing the voyage time, minimizing fuel consumption, and maximizing the safety factor. In this literature review section, selected works related to ice navigation are discussed. The purpose of this chapter is to provide a big picture of what has been done and the current gaps in previous research. Some typical researchers in pathfinding for ships in ice are Frederking (2003); Kotovirta et al. (2009); Choi et al. (2013); Choi et al. (2015); Liu et al. (2016); Piehl et al. (2017); and Lehtola et al. (2019). There are three main elements in a specific optimization problem: objectives, constraints, and optimization techniques. Therefore, the structure of this review section is organized by these three factors.

### 2. Objectives

The first element of optimization problems is objectives. Almost all research approaches the problem with shared objectives, including distance, time, and safety. The distance goal is simple, and safety is guaranteed by avoiding obstacles (land, island) and thick ice regimes. The time is determined by the inverse of the speed that the ship can achieve on each part of the route. This implementation makes sure the selected route can achieve the shortest time. Some research takes fuel consumption into account to make the problem more practical (Frederking, 2003; Piehl et al., 2017). Fuel consumption depends on two main factors: the speed of the vessel and the ice thickness of the environment (Frederking, 2003). The fuel amount is also related to the distance and time travelled. It is clear that fuel consumption, distance, and time objectives are conflicting. There is typically no route that the ship can have the shortest voyage in length in the shortest time and consume the smallest amount of fuel. However, none of the previous work mentions a trade-off between these objectives. In reality, the ship might take more time to travel on a longer route but use less fuel, and this route can become the optimal route if fuel saving is the first priority. In this research, we will investigate all of these practical objectives and identify the trade-offs between them.  Moreover, the other work considers smoothness in operational control of the ship, i.e. the average steering angle is not too sharp (Choi et al., 2013). This objective is only helpful when planning for a small area. The smoothness value is not essential where the granularity is usually greater than about one nautical mile. Therefore, this research ignores the smoothness to make the framework less complicated.

### 3. Constraints

The constraints of the pathfinding problem on ice have a significant impact on the practicality of the result. Some research differentiates between which ice-covered areas are navigable and which areas are not by using previous research guidelines for ice numeral calculation, such as Liu et al. (2016) and Choi et al. (2015). Lehtola et al. (2019) set up safety preferences to avoid the risk in a specific area. However, these rules are not standardized to apply to multiple regions. The other works ignore this constraint and assume that the ship can travel through all ice areas. This assumption is not valid and causes the result to be less reliable because each ship has an ice-class, and the ice capability of each ice class is different (IMO, 2016). The international and local regulatory constraints should be presented and enforced during navigation in ice conditions. These regulations are crucial because they provide guidelines to enhance safety at sea. All of the requirements must be executed strictly in reality. This research will attempt to solve the problem under an international constraint introduced by the International Maritime Organization, called the Polar Operational Limit Assessment Risk Indexing System (POLARIS) (IMO, 2016). POLARIS is a risk-based methodology to assess the operational limitations of vessels navigating in sea ice. This regulation entered into force in 2017.

## 4. Optimization method

The techniques of optimization problems are indispensable. There are many popular pathfinding algorithms in this area, such as graph-based techniques, genetic algorithms, and reinforcement learning. The rest of this chapter discusses these approaches.

The first approach is classified as the manual technique. Frederking (2003) is an example. He uses engineering techniques to analyze and compare some routes, then selects the best one. In the research, Frederking (2003) calculates the distance, time, and fuel consumption of the ship MV Cicero travelling through two different routes from Quebec City to Stephenville, Newfoundland. His approach breaks down the long route into small segments by their ice regimes. The distance, voyage time, and fuel consumption are calculated according to each regime's ice structure. Finally, all of them are summed, and each objective is compared one by one. The advantage of this approach is that it is straightforward. However, the disadvantage is that it is required to select some routes in advance for comparison. Furthermore, the method also has a problem when the ship type is changed. If other ships use this method, the lengthy manual process must be performed again because all of the calculations are only valid for the ship MV Cicero.

The second approach is based on the genetic algorithm (Goldberg, 1989), used in Choi et al. (2013). According to Goldberg (1989) and Choi et al. (2013), the method adopts the evolution theory in biology with chromosomes and several typical operations such as crossover, mutation, reproduction, random immigrant, and deletion. A chromosome represents an entire route, whose gene structure contains the position of the ship and some other properties, such as speed. At first, it creates a random generation and checks if the termination condition is met. If the condition is true, the process stops and results in the route. Otherwise, it regenerates the other solutions with some particular operations to change the chromosomes' structure until the termination point is reached. This approach's advantage is that it can be applied for both discrete and continuous maps, which makes the approach outstanding compared to other methods because they must discretize the map into grid cells. Nevertheless, the big challenge is that this algorithm's runtime is long, and the parameter setting of the system needs to be selected properly (Choi et al., 2013).

The third algorithm uses the graph search, such as Dijkstra (Dijkstra, 1959) and A* (Hart et al., 1968). This algorithm is applied by Choi et al. (2015), Liu et al. (2016), and Lehtola et al. (2019). In the graph, there are multiple vertices connected by some edges. One of the vertices is the origin port, and the other is the destination. Each edge has a value, showing a cost to go from one end to the other end. The cost could be time, distance, or the weighted summation of time and distance. The optimal route is the one whose summation of all the costs is the lowest. While Dijkstra’s method is the basic version, A* is an upgraded one by applying heuristics to speed up the convergence. Overall, this method's advantage is that it is easy to implement and is popular in pathfinding problems in general. However, the limitation of the simple graph is that the cost from one node to the other node in the graph is a fixed value, and the navigation is only accurate when the decision at each time step is the direction. The navigator may make the decision by changing many setpoints in the control system to drive a ship, such as direction (steering angle) and speed setpoint. In reality, the cost of going from point A to point B might have multiple values depending on what actions are taken. For example, the distance between A and B remains unchanged, but the time and fuel consumption will change if the ship runs at a different speed. This research will use a multigraph representation for the problem when two vertices can be linked by several edges.

In the literature, there are many other mathematical techniques used in this research area, such as (Kotovirta et al., 2009) and (Piehl et al., 2017). They create the models and formulate a cost function for the problem. Then they use mathematical methods to minimize the cost function, for example, the finite element method (Piehl et al., 2017) and Powell's method (Kotovirta et al., 2009). These methods might be exact and have a promising result for a specific problem. However, they are complicated and difficult to scale up. These solutions are also hard to apply to changing environments because the cost functions work properly only on given conditions.

Besides all the aforementioned techniques, reinforcement learning is an effective approach to solving pathfinding problems with many conflicting goals. It is the process of interaction between an agent and an environment. Interactive learning helps the agent to accumulate knowledge when the environment changes. In other words, reinforcement learning can solve the pathfinding problem under uncertainty. Many new reinforcement algorithms have been developed to address the multi-objective optimization problems (Van Moffaert et al., 2013; Van Moffaert & Nowé, 2014; Tozer et al., 2017). These theoretical approaches have not been applied to identify optimal routes for ships in ice. The differences between a single-objective problem and a multiple objective one are the reward signal and the way the agent selects actions. In the former context, the reward signal is scalar. It is feasible for the agent to pick the action with the highest score as the optimal choice. However, the reward in the multiple objective optimization problems is a vector, where each element represents an individual goal. Determination of the optimal action by comparing vectors to vectors becomes complicated in this situation because every single objective has a different range and unit. The first way to execute this comparison is using scalarization (Van Moffaert et al., 2013). This operation transforms a vector into a scalar value so that the agent can select actions easily. The advantage of the method is simplicity. The algorithm makes sure that there is always an optimal policy for a specific problem. When the coefficients of the scalarization vary, the optimal solution changes accordingly. The disadvantage is that it requires the user to choose a good setting in advance to have an expected result. The second approach to solve the multi-objective optimization problem is using multi-policy reinforcement learning, such as Pareto Q-learning (Van Moffaert & Nowé, 2014) and Voting Q-learning (Tozer et al., 2017). Van Moffaert & Nowé (2014) apply Pareto dominance relation to compare the reward vectors of possible actions. A vector X dominates a vector Y if and only if there exists one element of X that dominates the corresponding value of Y, and other elements in X are not dominated by their counterparts in Y (Censor, 1977). On the other hand, Tozer et al. (2017) use the social choice theory with various voting methods to compare the reward signals. Both techniques are advantageous to search for all possible optimal solutions for the problem. Nevertheless, the method's results cause trouble for the decision-makers. In the ice navigation situation, the random choice among the optimal set will lead to inconsistency. For these reasons, this research uses the simple scalarization Q-learning of reinforcement learning approach to apply for the pathfinding problems in ice-covered waters to validate the concept because of its straightforwardness.

## 5. Summary

This section reviews some selective works on pathfinding for ice-class vessels. The similarities and differences are discussed together with the favourable outcomes and downsides of each method. Although all previous work in this area provides models to identify a good route, the result might not be optimal. This research explores a conceptual framework to identify an optimal route for ice-class vessels using graph search and reinforcement learning, which follows the POLARIS guidelines.

# Chapter 3: Framework and Methods

## 1. Framework

This research attempts to build a general framework to resolve the route optimization problem for vessels through ice-covered waters. The framework proposed is an end-to-end solution (Figure 1). The early work is presented by Tran et al. (2020). The system's input is an ice chart, together with a start point and an endpoint in the map. A route selection algorithm is executed to generate an optimal route before a validation step checks whether this route meets all constraints. The output of the framework is the optimal route suggestion. This chapter discusses all elements in the model, including ice charts, route illustration, operational cost calculation by time and fuel consumption, and the POLARIS constraints.

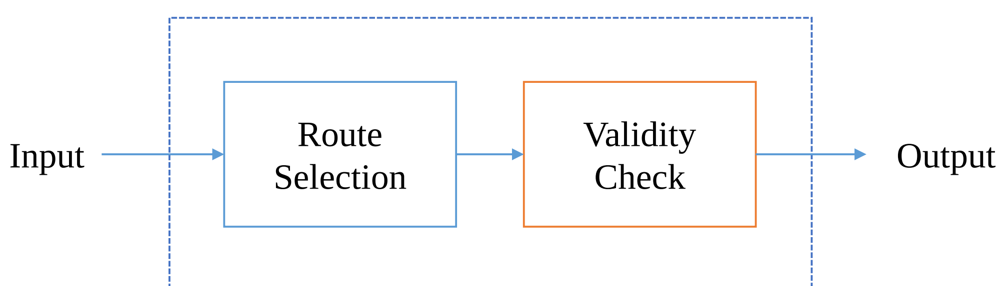


Figure 1. Overall framework (Tran et al., 2020)

**Ice chart:**

An ice chart is an estimation of ice for a specific time frame. In Canada, ice charts are issued by Canadian Ice Service (CIS). Depending on the region and the time of the year, the ice chart update might be on a daily or weekly basis. Figure 3 shows an example of an ice chart including multiple ice regimes, illustrated by egg codes. Each egg code shows the ice types and their concentrations. The details of how to interpret the egg code are from Canada Ice Service (2016). In general, it has four rows. The first row shows the total concentration of the ice regime. The next two rows list the contribution and the code of each ice type in the regime. The flow size indicators are in the last row. The decoding of the ice type code can be found in Table 1. Figure 2 illustrates an example of an ice egg code. The ice occupies approximately ten-tenths of the water in this regime, where thin first-year ice (code 7) takes up 4/10, the grey-white ice (code 5) and medium first-year ice (code 1·) share equally with 3/10 each.

Table 1. Coding of ice types (Canadian Ice Service, 2016)

| **Description** | **Thickness** | **Code** |
|---|---|---|
| New ice | < 10 centimetres | 1 |
| Nilas, Ice rind | < 10 centimetres | 2 |
| Young Ice | 10 - 30 centimetres | 3 |
| Grey Ice | 10 - 15 centimetres | 4 |
| Grey-white ice | 15 - 30 centimetres | 5 |
| First-year ice | >= 30 centimetres | 6 |
| Thin first-year ice | 30 - 70 centimetres | 7 |
| First stage thin first-year | 30 - 50 centimetres | 8 |
| Second stage thin first-year | 50 - 70 centimetres | 9 |
| Medium first-year ice | 70 - 120 centimetres | 1· |
| Thick first-year ice | > 120 centimetres | 4· |
| Old ice | - | 7· |
| Second-year ice | - | 8· |
| Multi-year ice | - | 9· |
| Ice of land origin | - | Δ· |
| Undetermined/Unknown | - | X· |

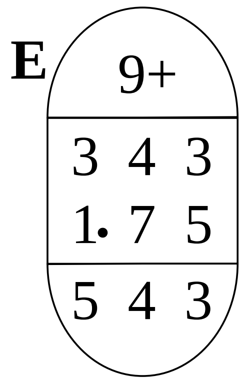


Figure 2. An example of an ice regime

The original ice chart is discretized into a grid, whose resolution relies on how detailed the user wants to plan the routes. In this conceptual model, the discretization of the ice chart is done manually. For instance, Figure 3 displays an ice chart of Newfoundland waters on March 11, 2020 and Figure 4 illustrates its discretized version.

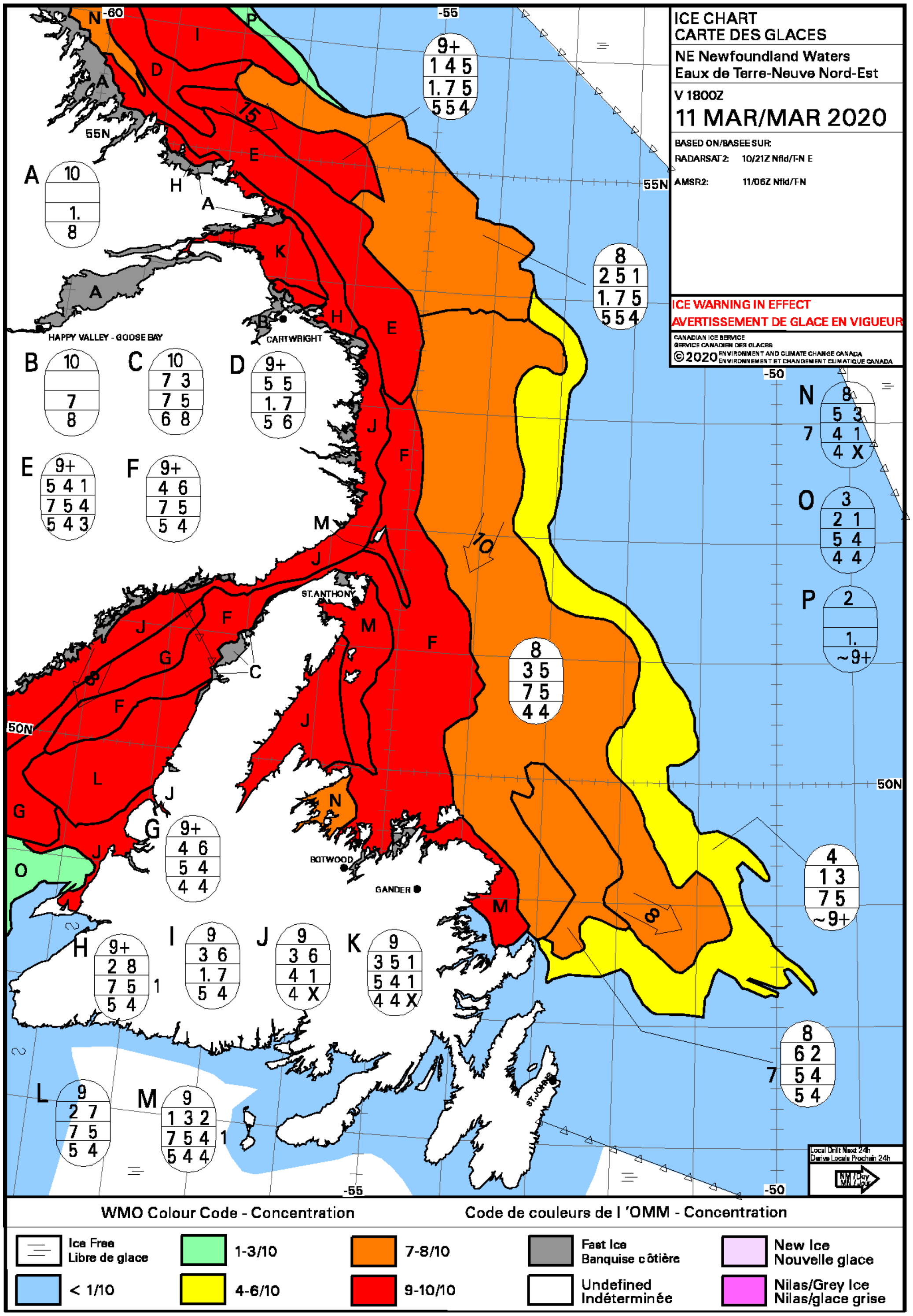


Figure 3. An ice chart of Newfoundland waters on March 11, 2020 (Ice Archive, 2019)

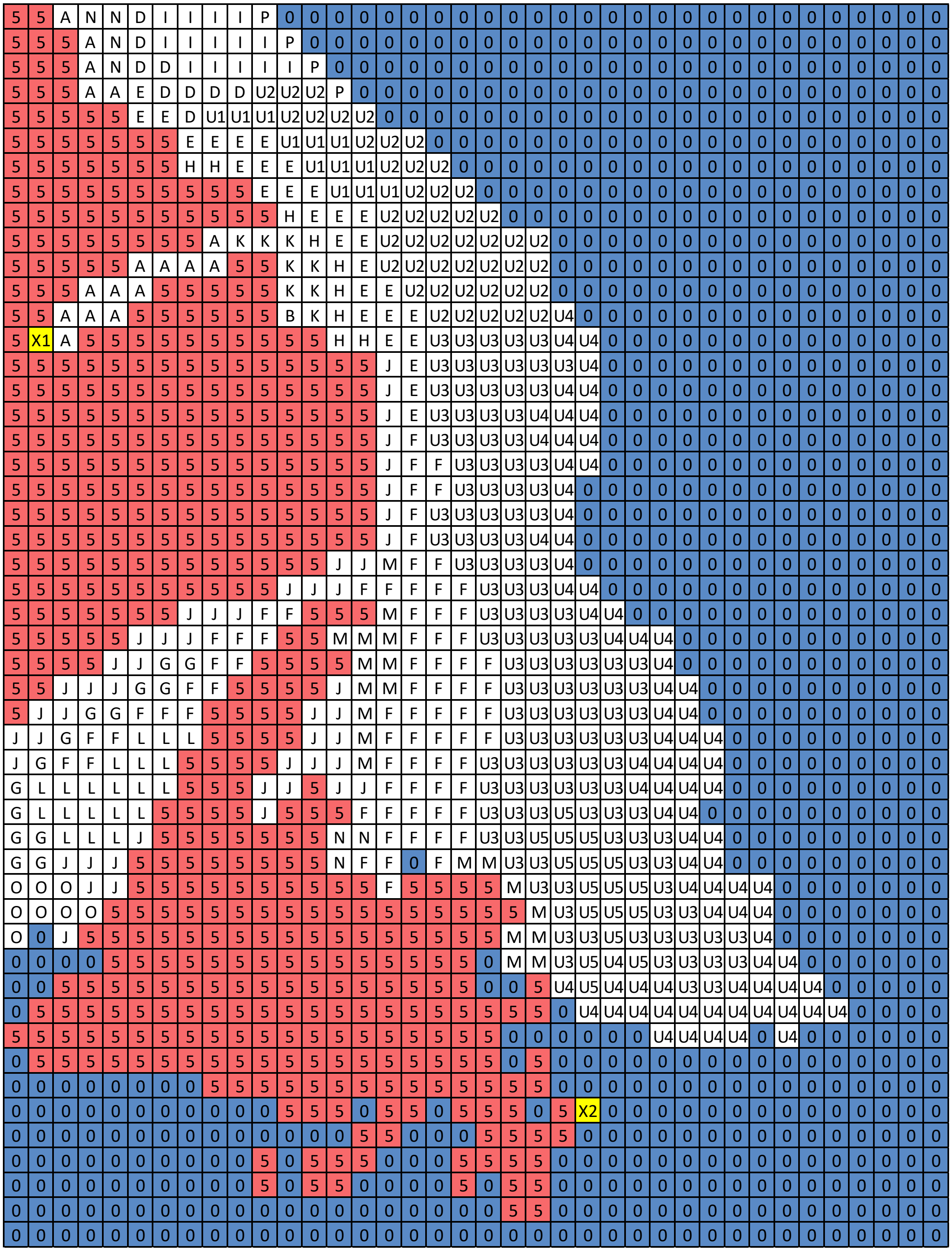

Figure 4. The discretized ice chart of Newfoundland waters on March 11, 2020

**Route:**

The route selection algorithm is the core of this framework. The route for navigation includes the direction and the speed along the way. In reality, the direction and speed of a ship should be in continuous domains, but in the scope of a conceptual framework, these factors are converted into discrete domains with eight directions and four speeds. Specifically, the direction includes north, northeast, east, southeast, south, southwest, west, northwest, and four speeds are 3, 5, 7, and 10 knots. In terms of route selection, two different methods are investigated: reinforcement learning and graph-based algorithms. The following section will discuss these approaches more.

**Operational indicators:**

The ultimate goal of the problem is to find the route that optimizes multiple competing objectives: the distance travelled, voyage time, and fuel consumption. It can be seen that all measurements are directly relevant to the operational cost. The lower the cost is, the better the route is. Moreover, fuel consumption is an essential factor because it has a significant impact on the natural environment. Reducing carbon emission is of importance in the system.

While the determination of the voyage time (units hour) is distance divided by speed, the resistance R, thrust T, power P, and fuel consumption are calculated by Equations (1)-(7). All equations used in the conceptual framework are simplified, based on the work of

Keinonen et al. (1996) and Frederking (2003) for the ship MV Cicero. This research assumes that the ship engines have unlimited power. The application for other ships might have different fuel functions. In this case, these equations can be replaced but there is no impact on the generality of the current framework. The fuel consumption in this work depends on the vessel speed V and ice thickness h. Note that speed is converted to m/s for calculation in these equations. The units of resistance and thrust are MN, while power and fuel consumption are measured in MW and tonnes.

$$\mathrm{R_{ice}} = 0.9\mathrm{h}^{1.5} \tag{1}$$

$$\mathrm{R_{ice}}(\mathrm{V} > 1\,{}^{\mathrm{m}}\!/_{\mathrm{s}}) = 0.138\mathrm{h}(\mathrm{V} - 1) \tag{2}$$

$$\mathrm{R_{OW}} = 0.011\mathrm{V} + 7.06 \times 10^{-6}\mathrm{V}^{5} \tag{3}$$

$$\mathrm{R_{Total}} = \mathrm{R_{ice}} + \mathrm{R_{ice}}(\mathrm{V} > 1\,{}^{\mathrm{m}}\!/_{\mathrm{s}}) + \mathrm{R_{OW}} \tag{4}$$

$$\mathrm{T} = \mathrm{R_{Total}} \tag{5}$$

$$\mathrm{P} = \frac{\mathrm{T}}{0.75 \times (0.122 - 0.0057\,\mathrm{V})} \tag{6}$$

$$\text{Fuel consumption} = 0.17 \times \mathrm{P} \times \text{time} \tag{7}$$

The calculation of the operation indicators in a regime is straightforward. An ice regime is usually a combination of multiple ice types with different ice concentrations (per tenth). The speed is a constant value in a cell. The fuel consumption to travel through an ice regime is the weighted summation of all fuel needed for going through each individual ice type of the regime. For example, let us assume an ice regime with an egg code, as in Figure 2. The speed is 5 knots. The travelled distance is 10 nautical miles (NM). With that information, the voyage time is 2 hours. The regime has three different ice types. The concentration per tenth of each type is equivalent to the length of distance that the ship sails through this type of ice. The individual and total fuel consumption are shown in Table 2.

Table 2. An example of fuel consumption on an ice regime

| | Thickness (m) | Ice Concentration | V (knots) | V (m/s) | Thrust (MN) | Power (MW) | Distance (NM) | Time (h) | Fuel (tonnes) |
|---|---|---|---|---|---|---|---|---|---|
| **ice type 1** | 1.2 | 0.3 | 5 | 2.58 | 1.47 | 17.01 | 3.00 | 0.60 | 1.73 |
| **ice type 2** | 0.7 | 0.4 | 5 | 2.58 | 0.71 | 8.18 | 4.00 | 0.80 | 1.11 |
| **ice type 3** | 0.3 | 0.3 | 5 | 2.58 | 0.24 | 2.80 | 3.00 | 0.60 | 0.29 |
| **open water** | 0 | 0 | 5 | 2.58 | 0.03 | 0.34 | 0.00 | 0.00 | 0.00 |
| **Total** | | | | | | | **10.00** | **2.00** | **3.13** |

## POLARIS Constraints

The constraint of route planning in the Arctic area is that the operation must follow the POLARIS guidelines. It imposes the operational constraints for all categories of ships by ice class in scenarios of ice conditions. This regulation was enforced in the Arctic region in 2017. The POLARIS constraints help ensure the safety of the ships when operating in ice-covered water.

In terms of the environment, waters are divided into multiple areas, including open water and ice regimes. Each regime has different characteristics such as ice type and ice thickness that are illustrated as an egg code in the ice chart. All the information about ice in this project is from Canadian Ice Service charts. Regarding the ice classes, ships are categorized a class by their ice capability, such as Polar Class 1 (PC1- highest), Polar Class 2 (PC2), down to ice-strengthened classes such as IC.

The POLARIS introduces a measurement called Risk Index Outcome (RIO), which is calculated by Risk Index Values (RIV) of each ice type in an egg code as Equation (8). The range of RIV is from -8 to 3. The higher the RIO is, the less risky the environment is.

$$\mathrm{RIO} = (\mathrm{C1} \times \mathrm{RIV1}) + (\mathrm{C2} \times \mathrm{RIV2}) + \ldots + (\mathrm{Cn} \times \mathrm{RIVn}) \tag{8}$$

where $C_i$ is the concentration of ice type i within the ice regimes, and $RIV_i$ is the corresponding RIV for ice type i.

Table 3 shows the RIV of different ice types and ice classes from the POLARIS. However, some stages of ice development from the CIS ice chart information do not match with this table. The current research proposes a modified RIV table for RIO calculation when the ice chart is from Canadian Ice Service (Table 4).

Table 3. Risk index values of different ice types and ice classes (IMO, 2016)

| | PC1 | PC2 | PC3 | PC4 | PC5 | PC6 | PC7 | IA Super | IA | IB | IC | Not Ice Strengthened |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Ice-Free** | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| **New Ice** | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 1 |
| **Grey Ice** | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 1 | 0 |
| **Grey White Ice** | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 1 | 0 | -1 |
| **Thin 1st Year ice 1st Stage** | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 1 | 0 | -1 | -2 |
| **Thin 1st Year Ice 2nd Stage** | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 0 | -1 | -2 | -3 |
| **Medium 1st Year Ice less than 1 m thick** | 2 | 2 | 2 | 2 | 1 | 1 | 0 | 0 | -1 | -2 | -3 | -4 |
| **Medium 1st Year Ice** | 2 | 2 | 2 | 2 | 1 | 0 | -1 | -1 | -2 | -3 | -4 | -5 |
| **Thick 1st Year Ice** | 2 | 2 | 2 | 1 | 0 | -1 | -2 | -2 | -3 | -4 | -5 | -6 |
| **Second Year Ice** | 2 | 1 | 1 | 0 | -1 | -2 | -3 | -3 | -4 | -5 | -6 | -7 |
| **Light Multi Year Ice less than 2.5 m thick** | 1 | 1 | 0 | -1 | -2 | -3 | -3 | -4 | -5 | -6 | -7 | -8 |
| **Heavy Multi Year Ice** | 1 | 0 | -1 | -2 | -2 | -3 | -3 | -4 | -5 | -6 | -8 | -8 |

Table 4. A proposed RIV table is used for the CIS ice chart

| Canadian Ice Code | POLARIS | PC1 | PC2 | PC3 | PC4 | PC5 | PC6 | PC7 | IA Super | IA | IB | IC | Not Ice Strengthened |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Ice-Free** | Ice-Free | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| **1** | New Ice | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 1 |
| **2** | | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 1 |
| **3** | | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 1 | 0 | -1 |
| **4** | Grey Ice | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 1 | 0 |
| **5** | Grey White Ice | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 1 | 0 | -1 |
| **6** | | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 0 | -1 | -2 | -3 |
| **7** | | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 0 | -1 | -2 | -3 |
| **8** | Thin First Year ice 1st Stage | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 1 | 0 | -1 | -2 |
| **9** | Thin First Year Ice 2nd Stage | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 0 | -1 | -2 | -3 |
| **1.** | Medium First Year Ice | 2 | 2 | 2 | 2 | 1 | 0 | -1 | -1 | -2 | -3 | -4 | -5 |
| **4.** | Thick First Year Ice | 2 | 2 | 2 | 1 | 0 | -1 | -2 | -2 | -3 | -4 | -5 | -6 |
| **7.** | | 1 | 1 | 0 | -1 | -2 | -3 | -3 | -4 | -5 | -6 | -7 | -8 |
| **8.** | Second Year Ice | 2 | 1 | 1 | 0 | -1 | -2 | -3 | -3 | -4 | -5 | -6 | -7 |

| Canadian Ice Code | POLARIS | PC1 | PC2 | PC3 | PC4 | PC5 | PC6 | PC7 | IA Super | IA | IB | IC | Not Ice Strengthened |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **9.** | Light Multi Year Ice | 1 | 1 | 0 | -1 | -2 | -3 | -3 | -4 | -5 | -6 | -7 | -8 |
| ∆ | Heavy Multi Year Ice | 1 | 0 | -1 | -2 | -2 | -3 | -3 | -4 | -5 | -6 | -8 | -8 |

The constraint of POLARIS for an ice-class vessel is in Table 5. Based on the ice-class and the RIO, the ship is allowed to operate normally, or with restrictions, or not at all. Table 6 indicates the suggested speeds for all ice classes with elevated operational risk.

Table 5. Operational criteria imposed by POLARIS (IMO, 2016)

| RIO | Ice classes PC1 – PC7 | Ice classes below PC7, or non-ice class |
|---|---|---|
| RIO $\geq$ 0 | Normal operation | Normal operation |
| -10 $\leq$ RIO < 0 | Elevated operational risk | Operation subject to special consideration |
| RIO < -10 | Operation subject to special consideration | Operation subject to special consideration |

Table 6. Safe speed limits for elevated risk operations (IMO, 2016)

| **Ice class** | **Safe speed limit** |
| --- | --- |
| PC1 | 11 knots |
| PC2 | 8 knots |
| PC3 – PC5 | 5 knots |
| Below PC5 | 3 knots |

A calculation of RIO for an ice regime is demonstrated for an ice-class PC5 as below. In accordance with the ice regime in Figure 2 and Table 4, the RIO = $3\times1 + 4\times2 + 3\times3 = 20$. This data reveals that the ship PC5 has normal operation adhering to all POLARIS guidelines.

In summary, POLARIS provides the guidelines for vessels operating in the Arctic area. The constraints force the navigators to adhere to the appropriate operation at a specific ice condition for different vessel classes to guarantee safety.

## 2. Reinforcement learning

Reinforcement learning (RL) is a method to solve the pathfinding problem. In the reinforcement learning context, the agent is a ship exploring the environment to discover the optimal path from one port to another (Figure 5). The problem can be represented as a Markov Decision Process. The agent keeps doing the same task thousands of times. In the beginning, it has no data nor experience about the environment. Hence the route it takes is not good. However, after a certain level of learning, the agent masters its skills to find the optimal route in a particular area. The final result is the best solution. The optimality here is relative based on the definition and expectation of the system designers. The designer regulates a factor called the reward value to evaluate how good or how bad an action is. The values of the reward function are usually scalar in single-objective optimization problems and a vector where each element represents a goal in multiple objective optimization problems. In this research, the Q-learning algorithm is used to update the experience of the agent.

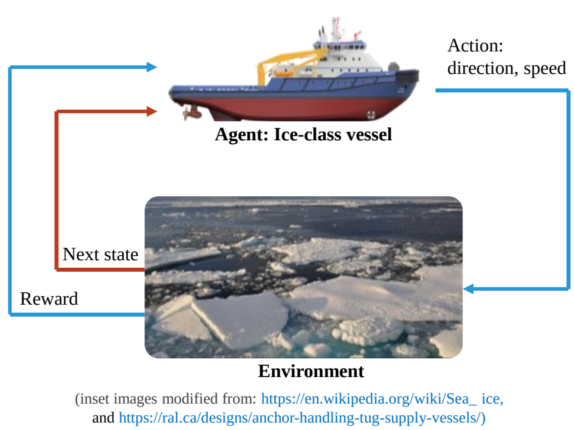


Figure 5. Reinforcement learning model (Tran et al., 2020)

**The agent:**

There are three different agents in our investigation, including PC5, PC7, and IC. The agents are named after the ship's classification. Each of them has different characteristics, especially hull strength. PC5 is the most ice-capable, PC7 is in the second rank, and IC has the lowest ice capability.

**The environment:**

The environment of the reinforcement learning system is a discretized version of an ice chart by the Canadian Ice Service. It is a grid world with m rows and n columns, where m and n vary according to resolutions. On the map, there is one departure port X1 and an arrival port X2. Other grid cells are represented by a character or 0 or 5. The characters are the egg codes of ice regimes, whereas 0 is open water, and 5 means land. An example of the environment is in Figure 4.

**Action:**

The agent takes a specific action at a time. The action includes two elements: direction and speed. In reality, the direction and speed of a ship should be in continuous domains, but in the scope of a conceptual framework, these factors are converted into discrete domains with eight directions and four speeds. Specifically, the direction includes north, northeast,

east, southeast, south, southwest, west, northwest, and four speeds are 3, 5, 7, and 10 knots. Therefore, there are 32 choices for the agent to select at a time. These actions are encoded by a number from 0 to 31, as the Table 7.

Table 7. List of actions in the reinforcement learning model

| Action code | Direction | Speed (knots) |
|---|---|---|
| **0** | North | 3 |
| **1** | North | 5 |
| **2** | North | 7 |
| **3** | North | 10 |
| **4** | Northeast | 3 |
| **5** | Northeast | 5 |
| **6** | Northeast | 7 |
| **7** | Northeast | 10 |
| **8** | East | 3 |
| **9** | East | 5 |
| **10** | East | 7 |
| **11** | East | 10 |
| **12** | Southeast | 3 |

| Action code | Direction | Speed (knots) |
|---|---|---|
| **13** | Southeast | 5 |
| **14** | Southeast | 7 |
| **15** | Southeast | 10 |
| **16** | South | 3 |
| **17** | South | 5 |
| **18** | South | 7 |
| **19** | South | 10 |
| **20** | Southwest | 3 |
| **21** | Southwest | 5 |
| **22** | Southwest | 7 |
| **23** | Southwest | 10 |
| **24** | West | 3 |
| **25** | West | 5 |
| **26** | West | 7 |
| **27** | West | 10 |
| **28** | Northwest | 3 |
| **29** | Northwest | 5 |
| **30** | Northwest | 7 |
| **31** | Northwest | 10 |

**State representation:**

Each state comprises the position of the agent in the environment, namely the coordinates (x,y).

**The reward function:**

The reward function in this system is a three-element vector [distance, time, fuel consumption].

$$reward\ r = \begin{cases} \left[-\mathrm{m}, -\frac{\mathrm{k}}{\mathrm{speed}}, -\mathrm{l} \times \mathrm{fuel}\right], \text{ for cardinal directions,} \\ \sqrt{2} \times \left[-\mathrm{m}, -\frac{\mathrm{k}}{\mathrm{speed}}, -\mathrm{l} \times \mathrm{fuel}\right], \text{ otherwise,} \end{cases} \tag{9}$$

where: m, k, l are parameters.

If the agent violates the POLARIS guideline, it receives a penalty tuple $[-\infty, -\infty, -\infty]$. There are three possible violations as below:

- IF RIO < -10
- IF (ice-class is PC3-PC7) AND (-10≤RIO<0) AND (speed > safe speed)
- IF (ice-class is IC) AND (RIO < 0)

**Q-learning:**

Q-learning is an algorithm in reinforcement learning (Sutton & Barto, 2018). Each pair of state S and action A has a value Q(S, A). Q-learning defines an update rule for all Q(S, A) when the agent chooses action A at state S. This algorithm aims to make sure that all Q values converge to their optimal values to have an optimal policy for the agent. The update rule follows Equation (10).

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)] \tag{10}$$

where: $\alpha$ is the learning rate,

$\gamma$ is the discount factor,

s and a are the current state and action, respectively,

s' and a' are the next state and action, respectively.

In Q-learning, the agent at state S has 32 possible actions. If it is in exploration mode, one random action is selected to go to the next state. Otherwise, the action having the highest Q value will be chosen. The exploration-exploitation trade-off is done by $\epsilon$-greedy. This approach requires the agent explore random actions with probability $\epsilon$, where $0 < \epsilon < 1$ (Sutton & Barto, 2018). Q values are vectors, not scalar. The comparison of vectors leads to another problem in the multi-objective optimization problem. To simplify, a linear scalarization method is used to convert the reward vector to a reward number. The score of the reward vector is determined by the summation of all elements of it. The advantage of this technique is that it is simple to implement and tune the weights to adjust the relationship between goals. However, it requires lots of work for the tuning process. Details of a scalarized Q-learning are in Algorithm 1.

**Algorithm 1**. The scalarized multiple objective Q-learning (Van Moffaert et al., 2013)

Initialize Q(s,a,o) arbitrarily

**For each** episode t **do**

    Initialize state s

    **Repeat**

        Choose action a from s using policy derived from Q-values (i.e. ε-greedy)

        Take action a and observe next state s' and reward r(s,a)

        **For each** objective o **do**

            $Q(s,a,o) \leftarrow Q(s,a,o) + \alpha[r(s,a,o) + \gamma \max_{a'} Q(s',a',o) - Q(s,a,o)]$

        **end for**

        $s \leftarrow s'$

    **until** s is terminal

**end for**

### 3. Graph-based method

Graph-based methods are popular in pathfinding problems. A graph G = (V, E) comprises a set of nodes V and a set of edges E connecting the nodes. In the discretized ice chart, each cell plays the role of one node in the graph. The cell is connected with its eight neighbours in eight directions: North, East, South, West, Northeast, Northwest, Southeast, Southwest. Each connectivity is similar to an edge with a value, which represents the cost to traverse from one node to another. Figure 6 shows the example of the graph representation in the pathfinding problem.

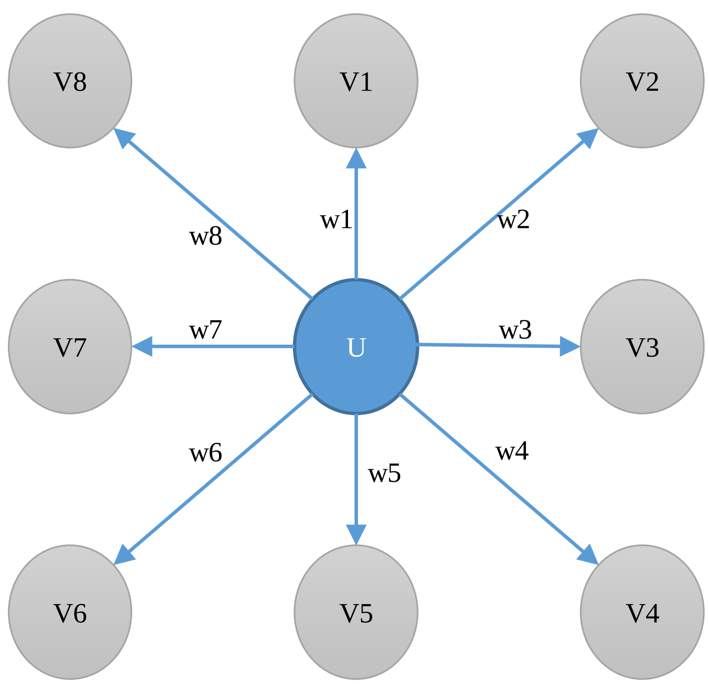


Figure 6. The connectivity at a node in the basic graph

However, the cost between two nodes remains unchanged in the basic graph, while the practical cost of moving between two nodes might vary when the speed changes. It requires the expansion of the basic graph to a multi-graph, where two nodes can be connected by many edges. Figure 7 shows an example of a multi-graph.

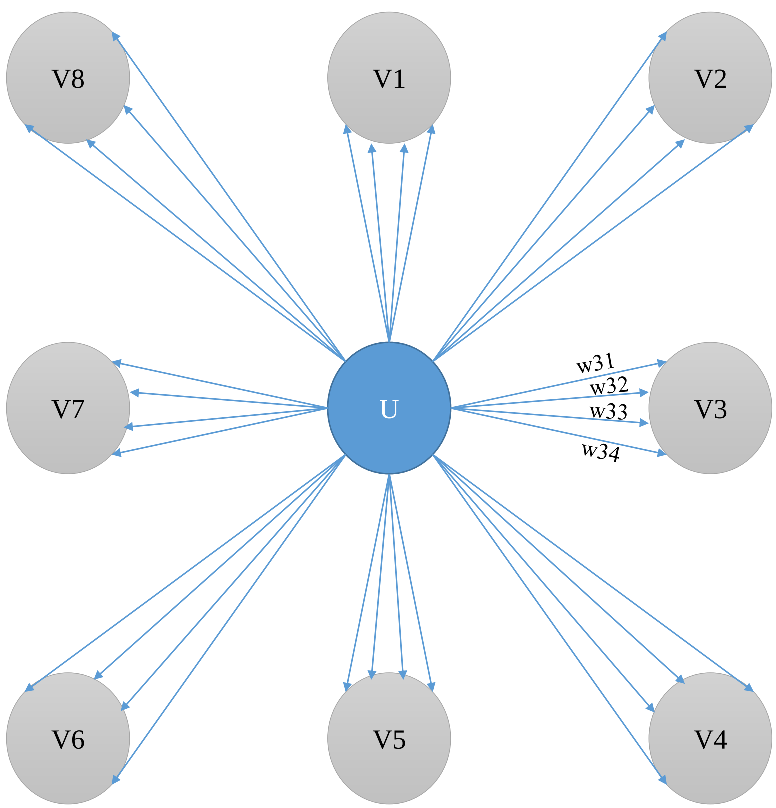


Figure 7. The connectivity at a node in multi-graph

The action of the ship has two elements: direction and speed. A multigraph approach is needed to handle this case. Each connectivity between two nodes U and V3 in Figure 7 has four edges instead of only one, representing four different speeds to travel from U to V3.

The cost function is set up the same as the reward signal but with an opposite sign. This is because the reinforcement learning goal is to maximize the reward signal, while that of the graph is to minimize the cost. For instance, if the reward value of an action taken from state s1 to state s2 is -20, the corresponding cost value is 20 in the graph search model.

There are two algorithms of graph-based methods considered in this work. They are Dijkstra's algorithm (Dijkstra, 1959) and A* (Hart et al., 1968). The difference is that Dijkstra's method tries to minimize the cost from the original point, while the A* minimizes the summation of the cost from the original point and the estimated cost to the endpoint. The advantage of A* algorithm is that it solves the problem faster than Dijkstra's algorithm. However, the approximation of the heuristic cost becomes a disadvantage of A* approach. If the estimation is not good, the runtime of the two algorithms is identical. In the current problem, the total cost comprises distance, time, and fuel consumption cost. While the distance from a point to the destination is estimated by the Manhattan or Euclidean method, predicting time and fuel consumption is more difficult because they depend on speed and ice conditions. This challenge is a pitfall to apply A* to solve the problem. On the other hand, Dijkstra's algorithm does not face this difficulty. Therefore, only Dijkstra's algorithm is used in this research.

The pseudo-code of the graph-based algorithms is Algorithm 2.

**Algorithm 2**: Dijkstra's algorithm (Cormen et al., 2009)

```
Q = G.V, a set of all vertices of the graph G
For each vertex v in Q:
        cost [v] = +∞            # The cost from each vertex to the departure is infinite
        source [v] = NULL
End for
cost [departure] = 0

While Q ≠ ∅:
        u = heapq.heappop(Q)          # select the vertex u with the minimum cost,
                                      # using priority queue and remove it out of Q
        For each vertex v that is linked to u:
                If cost [v] > cost [u] + w:
                        cost [v] = cost [u] + w
                        source [v] = u
                End if
        End for
End while
```

# Chapter 4: Trials and Results

This research demonstrates five investigations to show how the framework operates with two different approaches and how the performance compares with each other:

- The first trial deals with an idealized simple grid world environment to verify the concept.
- The second one extends the model with a simulation of uncertainty.
- The third case is for a real ice chart in a deterministic context.
- The expansion of this work is illustrated in the fourth part, where the trial considers the daily changes in the ice conditions.
- Finally, a validation with experts is performed to verify the results from this research.

## 1. Deterministic environment

In this trial, three agents, PC5, PC7, and IC, will find the optimal path from X1 to X2 in an idealized environment represented in Figure 8. In the ice chart, 0 and 5 represent open water and land, respectively, whereas the alphabet characters are ice regimes. The detail of the ice concentration of all ice types and the RIO of each regime is displayed in Table 8 and Table 9. The weights of three objectives are arbitrarily chosen, where m = 3.5, k = 1, l = 1. This evaluation aims to test the entire framework and check the behaviours of different ice classes for the same environment. The hyper-parameters of Q-learning are set as below, $\alpha = 1.0, \gamma = 1.0, \varepsilon = 0.1$. The number of training episodes is 2,000. The running time of two different approaches for three agents is visualized in Table 10.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | Q | Q | Q | Q | Q | Q | Q | Q | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | N | N | N | N | N | N | N | N | 0 | 0 |
| 0 | 0 | M | M | M | M | M | M | M | M | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | M | M | M | M | M | M | M | M | 0 | 0 |
| X1 | 0 | L | L | L | L | L | L | L | L | 0 | X2 |

Figure 8. A simulated ice chart

Table 8. The ice concentration of all ice types of the simulated ice chart (per tenth)

| Regime | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1dot | 4dot | 7dot | 8dot | 9dot | delta |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **L** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 6 | 0 | 0 |
| **M** | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 0 | 0 |
| **N** | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 8 | 0 | 0 | 0 | 0 | 0 |
| **P** | 0 | 0 | 2 | 3 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Q** | 0 | 3 | 3 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **R** | 0 | 0 | 8 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 9. The RIO of all ice regime of the simulated ice chart

| Regime | L | M | N | P | Q | R | 0 |
|---|---|---|---|---|---|---|---|
| **PC5** | -15 | -5 | 14 | 25 | 26 | 28 | 30 |
| **PC7** | -21 | -15 | -4 | 15 | 16 | 18 | 30 |
| **IC** | -54 | -45 | -30 | -3 | 5 | 6 | 30 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | Q | Q | Q | Q | Q | Q | Q | Q | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | N | N | N | N | N | N | N | N | 0 | 0 |
| 0 | 0 | M | M | M | M | M | M | M | M | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0-10 | M-5 | M-5 | M-5 | M-5 | M-5 | M-5 | M-5 | M-5 | 0-10 | 0 |
| X1 | 0 | L | L | L | L | L | L | L | L | 0 | X2-10 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | Q | Q | Q | Q | Q | Q | Q | Q | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | N | N | N | N | N | N | N | N | 0 | 0 |
| 0 | 0 | M | M | M | M | M | M | M | M | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0-10 | M-5 | M-5 | M-5 | M-5 | M-5 | M-5 | M-5 | M-5 | 0-10 | 0 |
| X1 | 0 | L | L | L | L | L | L | L | L | 0 | X2-10 |

Figure 9. The route suggested for PC5 (left: Dijkstra's algorithm, right: RL)

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | Q | Q | Q | Q | Q | Q | Q | Q | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | P | P-10 | P-10 | P-10 | P-10 | P-10 | P-10 | P | 0 | 0 |
| 0 | 0 | N-3 | N | N | N | N | N | N | N-3 | 0 | 0 |
| 0 | 0-10 | M | M | M | M | M | M | M | M | 0-10 | 0 |
| 0-10 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0-10 | 0 |
| 0-10 | 0 | M | M | M | M | M | M | M | M | 0-10 | 0 |
| X1 | 0 | L | L | L | L | L | L | L | L | 0 | X2-10 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | Q | Q | Q | Q | Q | Q | Q | Q | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | P | P-10 | P-10 | P-10 | P-10 | P-10 | P-10 | P | 0 | 0 |
| 0 | 0 | N-3 | N | N | N | N | N | N | N-3 | 0 | 0 |
| 0 | 0-10 | M | M | M | M | M | M | M | M | 0-10 | 0 |
| 0-10 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0-10 | 0 |
| 0-10 | 0 | M | M | M | M | M | M | M | M | 0-10 | 0 |
| X1 | 0 | L | L | L | L | L | L | L | L | 0 | X2-10 |

Figure 10. The route suggested for PC7 (left: Dijkstra's algorithm, right: RL)

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | Q-10 | Q-10 | Q-10 | Q-10 | Q-10 | Q-10 | Q-10 | Q-10 | 0 | 0 |
| 0 | 0-10 | P | P | P | P | P | P | P | P | 0-10 | 0 |
| 0 | 0-10 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0-10 | 0 |
| 0-10 | 0 | P | P | P | P | P | P | P | P | 0-10 | 0 |
| 0-10 | 0 | P | P | P | P | P | P | P | P | 0-10 | 0 |
| 0-10 | 0 | N | N | N | N | N | N | N | N | 0-10 | 0 |
| 0-10 | 0 | M | M | M | M | M | M | M | M | 0-10 | 0 |
| 0-10 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0-10 | 0 |
| 0-10 | 0 | M | M | M | M | M | M | M | M | 0-10 | 0 |
| X1 | 0 | L | L | L | L | L | L | L | L | 0 | X2-10 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | Q-10 | Q-10 | Q-10 | Q-10 | Q-10 | Q-10 | Q-10 | Q-10 | 0 | 0 |
| 0 | 0-10 | P | P | P | P | P | P | P | P | 0-10 | 0 |
| 0 | 0-10 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0-10 | 0 |
| 0-10 | 0 | P | P | P | P | P | P | P | P | 0-10 | 0 |
| 0-10 | 0 | P | P | P | P | P | P | P | P | 0-10 | 0 |
| 0-10 | 0 | N | N | N | N | N | N | N | N | 0-10 | 0 |
| 0-10 | 0 | M | M | M | M | M | M | M | M | 0-10 | 0 |
| 0-10 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0-10 | 0 |
| 0-10 | 0 | M | M | M | M | M | M | M | M | 0-10 | 0 |
| X1 | 0 | L | L | L | L | L | L | L | L | 0 | X2-10 |

Figure 11. The route suggested for IC (left: Dijkstra's algorithm, right: RL)

Table 10. Running time (seconds)

| Ship | Dijkstra's algorithm | Reinforcement learning |
|---|---|---|
| **PC5** | 0.047 | 11.932 |
| **PC7** | 0.047 | 14.486 |
| **IC** | 0.047 | 18.716 |

According to Table 9, POLARIS prevents PC5 from operating in regime L since RIO is less than -10. The ship has to slow down in regime M due to RIO at -5 while allowing operate normally in other regions. Similarly, PC7 cannot go in regimes L and M. The other regimes are permissible, except regime N with speed reduction. IC only has a green signal in regimes Q, R, and open water.

The results are shown in Figure 9, Figure 10, and Figure 11, where the routes comprise the regime code together with speed suggestions. In general, the results of both methods are the same and strictly abide by the rules. PC5 has the shortest route with a track through regime M. The maximum speed of the ship in these regimes is 5 knots to comply with the POLARIS when the regime has a negative RIO. Meanwhile, the PC7 chooses a longer route through regime P. Although the ship can operate in regime N to have a shorter route, it prefers going a little farther to run at full speed as the optimal decision than moving slowly at 3 knots in regime N. This result comes from the calibration of three objectives as mentioned earlier. Last but not least, IC has the longest route through regime Q because it is the least ice-capable ship.

### 2. Non-deterministic environment

While the first work has the assumption that the ice regimes are unchanged, the next considers a level of uncertainty in the ice chart prediction to make the problem more realistic. The ocean is a dynamic environment. Let us take ice drifting into consideration and assume that the change can only happen within a one-cell area. This means an ice regime can be replaced by any of its eight closest neighbours in all eight directions. Generally, a regime has only two possible statuses: affected by ice drifting with a probability p and remain unchanged, with a probability of 1-p, where $0 < p < 1$. When ice drifting happens, the ice regime is determined by the worst case of all possibilities to simplify the problem, in which the worst case has the smallest RIO.

An illustration of ice drift is shown in Figure 12. The current regime is open water. It is close to an ice regime N and ice regime M to the west and land to the southwest. Other neighbours are all open waters. According to the description above, this regime remains unchanged for 1-p of the visit, while it becomes ice regime M in another probability p of the time. This is because regime M is more severe than regime N regarding RIO, and the land is static, and its movement is not sensible. When $p = 0$, and $p = 1$, the environment becomes deterministic as Figure 13. Dijkstra's algorithm can only work with the unchanged (deterministic environment) graph. This trial will investigate the routes of Dijkstra's algorithm with p = 1 and that of Q-learning with p = 0.2. The hyper-parameters of Q-learning are set as below, $\alpha = 0.01, \gamma = 1.0, \varepsilon = 0.1$. The number of training episodes is 300,000. The running time of two different approaches for three agents is visualized in Table 11.

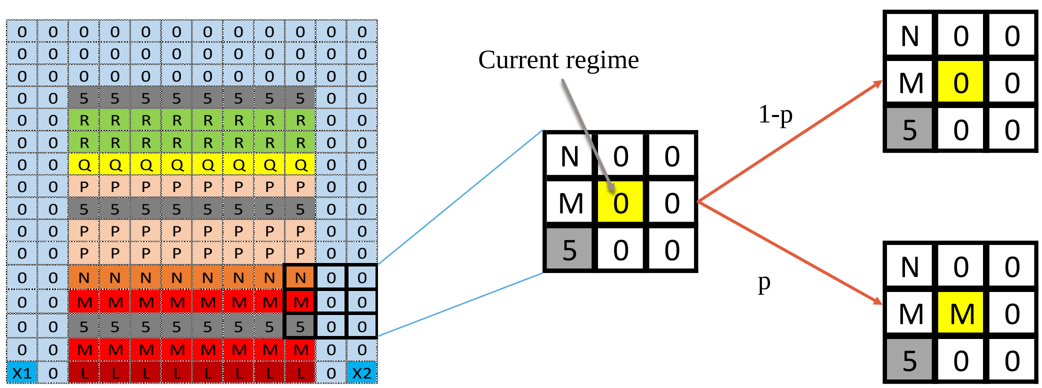


Figure 12. An example of ice drifting

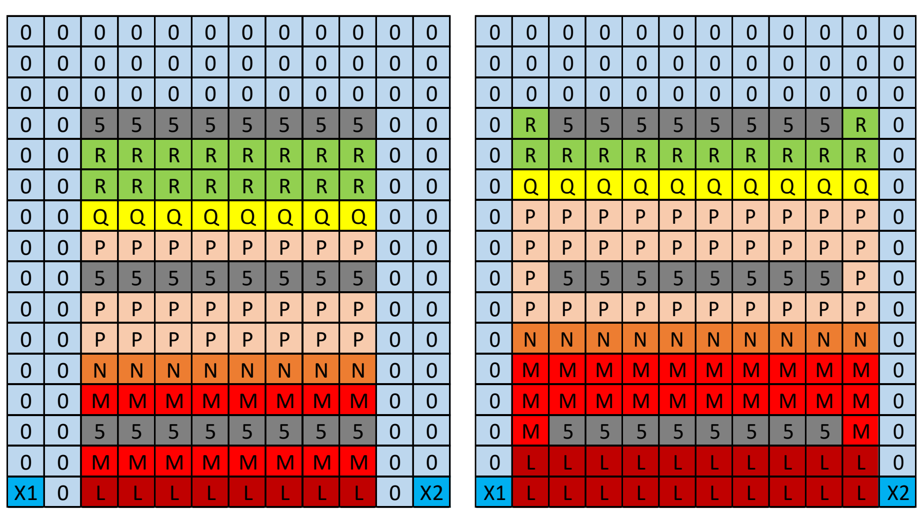

Figure 13. Example of environment when p = 0 (left), and p = 1 (right)

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | Q | Q | Q | Q | Q | Q | Q | Q | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | N | N | N | N | N | N | N | N | 0 | 0 |
| 0 | 0 | M-5 | M-5 | M-5 | M-5 | M-5 | M-5 | M-5 | M-5 | 0 | 0 |
| 0 | 0-5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0-5 | 0 |
| 0-10 | 0 | M | M | M | M | M | M | M | M | 0 | 0-10 |
| X1 | 0 | L | L | L | L | L | L | L | L | 0 | X2-10 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | Q | Q | Q | Q | Q | Q | Q | Q | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | N | N | N | N | N | N | N | N | 0 | 0 |
| 0 | 0 | M-5 | M-5 | M-5 | M-5 | M-5 | M-5 | M-5 | M-5 | 0 | 0 |
| 0 | 0-5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0-5 | 0 |
| 0-10 | 0 | M | M | M | M | M | M | M | M | 0 | 0-10 |
| X1 | 0 | L | L | L | L | L | L | L | L | 0 | X2-10 |

Figure 14. The route suggested for PC5 (left: Dijkstra's algorithm, right: RL)

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | Q | Q | Q | Q | Q | Q | Q | Q | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | P-10 | P-10 | P-10 | P-10 | P-10 | P-10 | P-10 | P-10 | 0 | 0 |
| 0 | 0-3 | P | P | P | P | P | P | P | P | 0-3 | 0 |
| 0-10 | 0 | N | N | N | N | N | N | N | N | 0 | 0-10 |
| 0-10 | 0 | M | M | M | M | M | M | M | M | 0 | 0-10 |
| 0-10 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0-10 |
| 0-10 | 0 | M | M | M | M | M | M | M | M | 0 | 0-10 |
| X1 | 0 | L | L | L | L | L | L | L | L | 0 | X2-10 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0 | Q | Q | Q | Q | Q | Q | Q | Q | 0 | 0 |
| 0 | 0 | P | P | P | P | P | P | P | P | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | P-10 | P-10 | P-10 | P-10 | P-10 | P-10 | P-10 | P-10 | 0 | 0 |
| 0 | 0-3 | P | P | P | P | P | P | P | P | 0-3 | 0 |
| 0-10 | 0 | N | N | N | N | N | N | N | N | 0 | 0-10 |
| 0-10 | 0 | M | M | M | M | M | M | M | M | 0 | 0-10 |
| 0-10 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0-10 |
| 0-10 | 0 | M | M | M | M | M | M | M | M | 0 | 0-10 |
| X1 | 0 | L | L | L | L | L | L | L | L | 0 | X2-10 |

Figure 15. The route suggested for PC7 (left: Dijkstra's algorithm, right: RL)

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0-10 | R-10 | R-10 | R-10 | R-10 | R-10 | R-10 | R-10 | R-10 | 0-10 | 0 |
| 0-10 | 0 | Q | Q | Q | Q | Q | Q | Q | Q | 0 | 0-10 |
| 0-10 | 0 | P | P | P | P | P | P | P | P | 0 | 0-10 |
| 0-10 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0-10 |
| 0-10 | 0 | P | P | P | P | P | P | P | P | 0 | 0-10 |
| 0-10 | 0 | P | P | P | P | P | P | P | P | 0 | 0-10 |
| 0-10 | 0 | N | N | N | N | N | N | N | N | 0 | 0-10 |
| 0-10 | 0 | M | M | M | M | M | M | M | M | 0 | 0-10 |
| 0-10 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0-10 |
| 0-10 | 0 | M | M | M | M | M | M | M | M | 0 | 0-10 |
| X1 | 0 | L | L | L | L | L | L | L | L | 0 | X2-10 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 |
| 0 | 0 | R | R | R | R | R | R | R | R | 0 | 0 |
| 0 | 0-10 | R-10 | R-10 | R-10 | R-10 | R-10 | R-10 | R-10 | R-10 | 0-10 | 0 |
| 0-10 | 0 | Q | Q | Q | Q | Q | Q | Q | Q | 0 | 0-10 |
| 0-10 | 0 | P | P | P | P | P | P | P | P | 0 | 0-10 |
| 0-10 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0-10 |
| 0-10 | 0 | P | P | P | P | P | P | P | P | 0 | 0-10 |
| 0-10 | 0 | P | P | P | P | P | P | P | P | 0 | 0-10 |
| 0-10 | 0 | N | N | N | N | N | N | N | N | 0 | 0-10 |
| 0-10 | 0 | M | M | M | M | M | M | M | M | 0 | 0-10 |
| 0-10 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0-10 |
| 0-10 | 0 | M | M | M | M | M | M | M | M | 0 | 0-10 |
| X1 | 0 | L | L | L | L | L | L | L | L | 0 | X2-10 |

Figure 16. The route suggested for IC (left: Dijkstra's algorithm, right: RL)

Table 11. Running time of the environment with ice drifting (seconds)

| Ship | Dijkstra's algorithm | Reinforcement learning |
|---|---|---|
| **PC5** | 0.047 | 1670 |
| **PC7** | 0.047 | 1753 |
| **IC** | 0.047 | 2356 |

The results are shown in Figure 14, Figure 15, and Figure 16 for PC5, PC7, and IC, respectively. The route of PC5 chooses to go north in the beginning before turning to regime M (the fourth row from bottom) and moving south to the destination at the right edge of the map. In the previous trial, this result is a significant change. The optimal result in trial 1 (Figure 9) is not valid anymore when the ice drifting exists. If the ship PC5 keeps moving in the positions near regime L, there is a chance that it hits the severe ice caused by the drifting of regime L. By interactive learning, the Q-learning agent knows to avoid this area. The selected route is optimized and in the safe zones.

PC7 in this trial also chooses a safer route than that of the first evaluation. As can be seen from Figure 10 and Figure 15, there are some differences between these routes. In the deterministic environment, the ship chooses the shortest route and optimizes the cost by skipping the severe ice in L and M regimes. However, the route suggestion in a non-deterministic case is to go north farther to keep staying one step away from the severe ice.

This is because the environment can change at any time based on the assumption of ice drifting, as described earlier. If the agent moves next to the edge of the more severe ice, there is a chance that the ice will drift around and cause more risk for the ship.

Similarly, IC changes the routes from going through regime Q (Figure 11) to regime R (Figure 16) as the optimal solution.

### 3. Route from Goose Bay to St. John's

After verification steps in the first two trials, this stage evaluates the framework with a realistic route finding problem. The third trial uses a realistic ice chart of the North East Newfoundland area. This chart was issued for ice estimation on March 11, 2020. The task is finding the route from Goose Bay (point A) to St. John's (point B), as shown in Figure 3. Figure 4 shows the discretized ice chart with 38×50 cells. Each cell is approximately a 12×12 square nautical mile area in reality. Both deterministic and non-deterministic cases (ice drift) are considered. As can be seen in Figure 3, there are 22 different regimes in the chart, including open water. The ice is severe along the coastal line and less risky offshore. The ice concentration of all ice regimes is shown in Table 12. The corresponding RIOs are also calculated in Table 13.

The aforementioned method states that the objective vector of distance, time, and fuel consumption is scalarized into a score. Every weight setting can come up with an optimal result. The question is how to nominate the outstanding results as the route decision among the optimal set. This section runs several combinations of weights of the three objectives as a grid search for calibration to address the concern. Three extreme cases are investigated first when each element of distance, time, and fuel becomes the sole goal for the optimization in the framework. This process helps determine the boundary of the metrics. The other settings are tested, and the results are compared to the above corner cases to come out with a reasonable route. The measurement of distance, time and fuel consumption of each route are displayed in Table 14.

Table 12. The ice concentration of all ice types of the Newfoundland ice chart

| Regime | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1dot | 4dot | 7dot | 8dot | 9dot | delta |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **A** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 | 0 |
| **B** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **C** | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **D** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 |
| **E** | 0 | 0 | 0 | 0 | 1 | 4 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **F** | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **G** | 0 | 0 | 0 | 0 | 6 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Regime | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1dot | 4dot | 7dot | 8dot | 9dot | delta |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **H** | 0 | 0 | 0 | 0 | 0 | 8 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **I** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 |
| **J** | 1 | 6 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **K** | 1 | 1 | 0 | 0 | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **L** | 1 | 0 | 0 | 0 | 0 | 7 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **M** | 4 | 0 | 0 | 0 | 2 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **N** | 2 | 3 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **O** | 7 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **P** | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 |
| **U1** | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| **U2** | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 5 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 |
| **U3** | 2 | 0 | 0 | 0 | 0 | 5 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **U4** | 6 | 0 | 0 | 0 | 0 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **U5** | 2 | 0 | 0 | 0 | 2 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 13. RIO calculation of ice regimes of the ice chart on March 11, 2020

| **Regime** | **0** | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** | **J** | **K** | **L** | **M** | **N** | **O** | **P** | **U1** | **U2** | **U3** | **U4** | **U5** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **PC5** | 30 | 10 | 20 | 23 | 15 | 25 | 26 | 30 | 28 | 18 | 30 | 30 | 28 | 29 | 30 | 30 | 26 | 24 | 21 | 27 | 29 | 30 |
| **PC7** | 30 | -10 | 10 | 13 | 0 | 15 | 16 | 20 | 18 | 6 | 21 | 21 | 19 | 23 | 22 | 27 | 22 | 13 | 11 | 19 | 25 | 22 |
| **IC** | 30 | -40 | -20 | -14 | -30 | -9 | -8 | 6 | -4 | -21 | 18 | 10 | -1 | 12 | 17 | 22 | 16 | -12 | -12 | 0 | 16 | 3 |

Table 14. Metrics of different routes for PC5 and PC7

| | **PC5** | | | **PC7** | | |
|---|---|---|---|---|---|---|
| **(m, k, l)** | distance (NM) | time (h) | fuel (tonnes) | distance (NM) | time (h) | fuel (tonnes) |
| **(0, 0, 1)** | 917 | 262 | **69** | 917 | 285 | **107** |
| **(0, 1, 0)** | 620 | **62** | 86 | 624 | **86** | 138 |
| **(1, 0, 0)** | **620** | 207 | 137 | **620** | 207 | 137 |
| **(1, 1, 0)** | 624 | 63 | 87 | 624 | 86 | 126 |
| **(1, 0, 1)** | 624 | 100 | 85 | 624 | 123 | 123 |
| **(0, 1, 1)** | 753 | 75 | 78 | 753 | 99 | 116 |
| **(1, 1, 1)** | 624 | 63 | 87 | 624 | 86 | 125 |
| **(0.1, 1, 4)** | **732** | **73** | **79** | **732** | **96** | **117** |

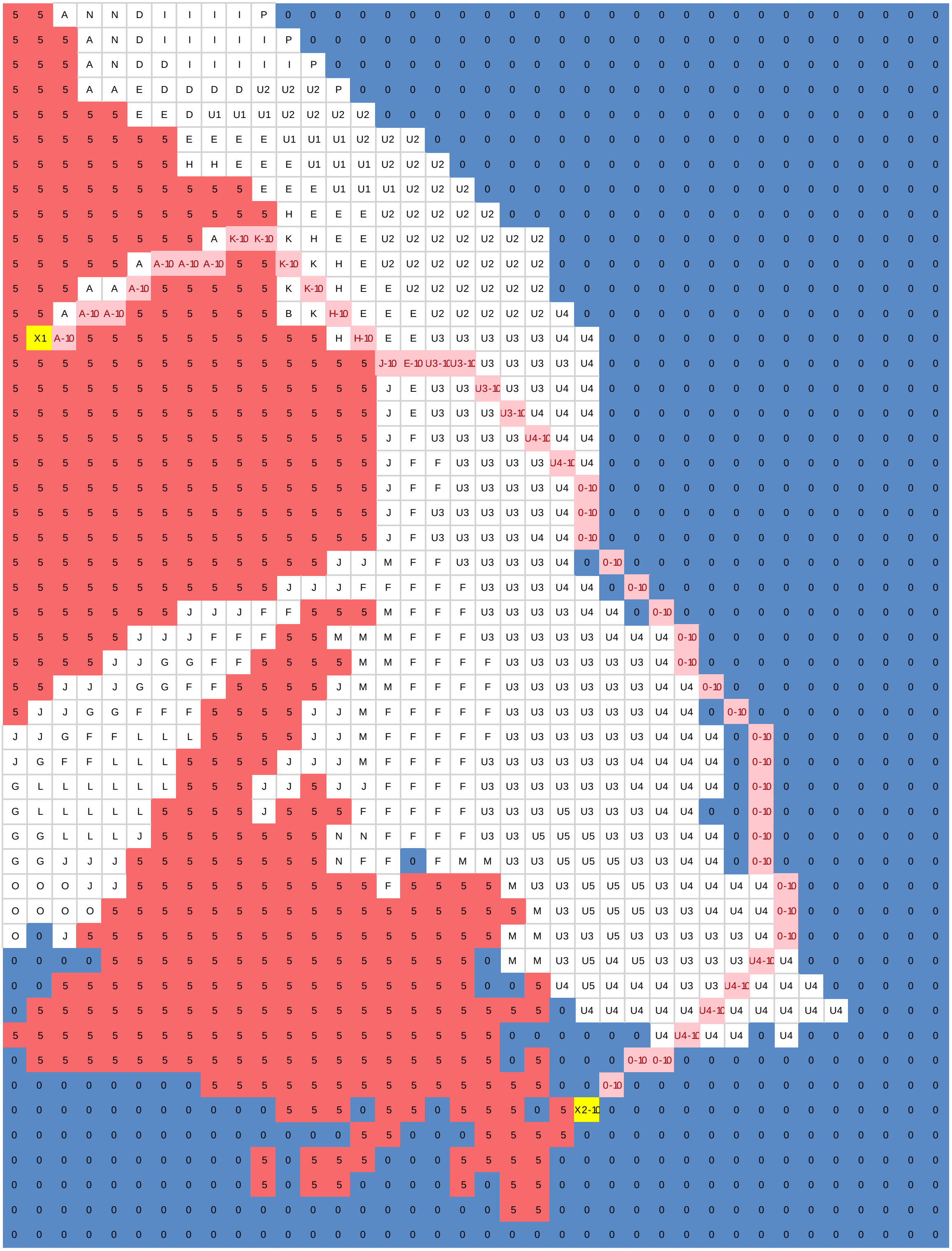

Figure 17. Route of PC5 in the deterministic environment

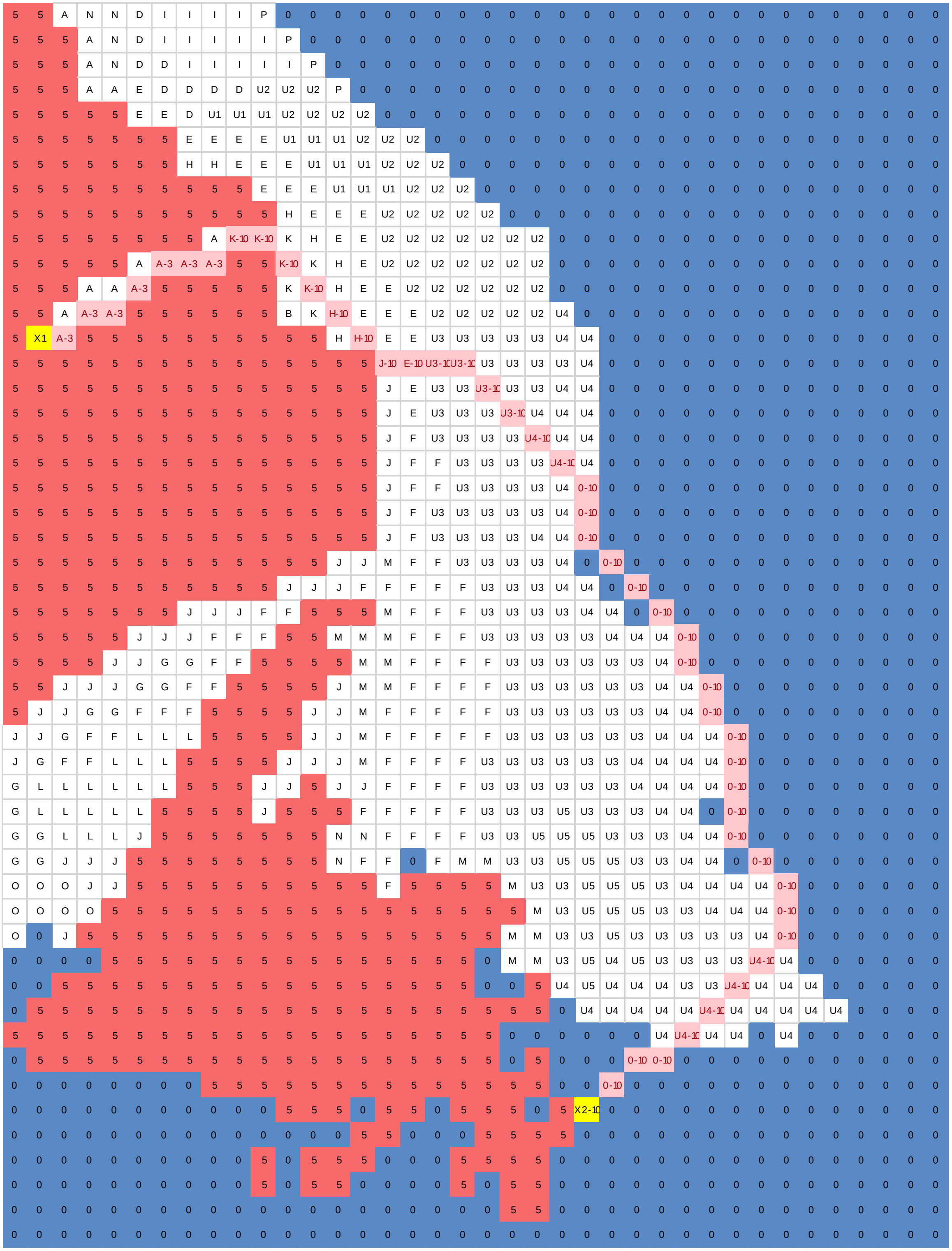

Figure 18. Route of PC7 in a deterministic environment

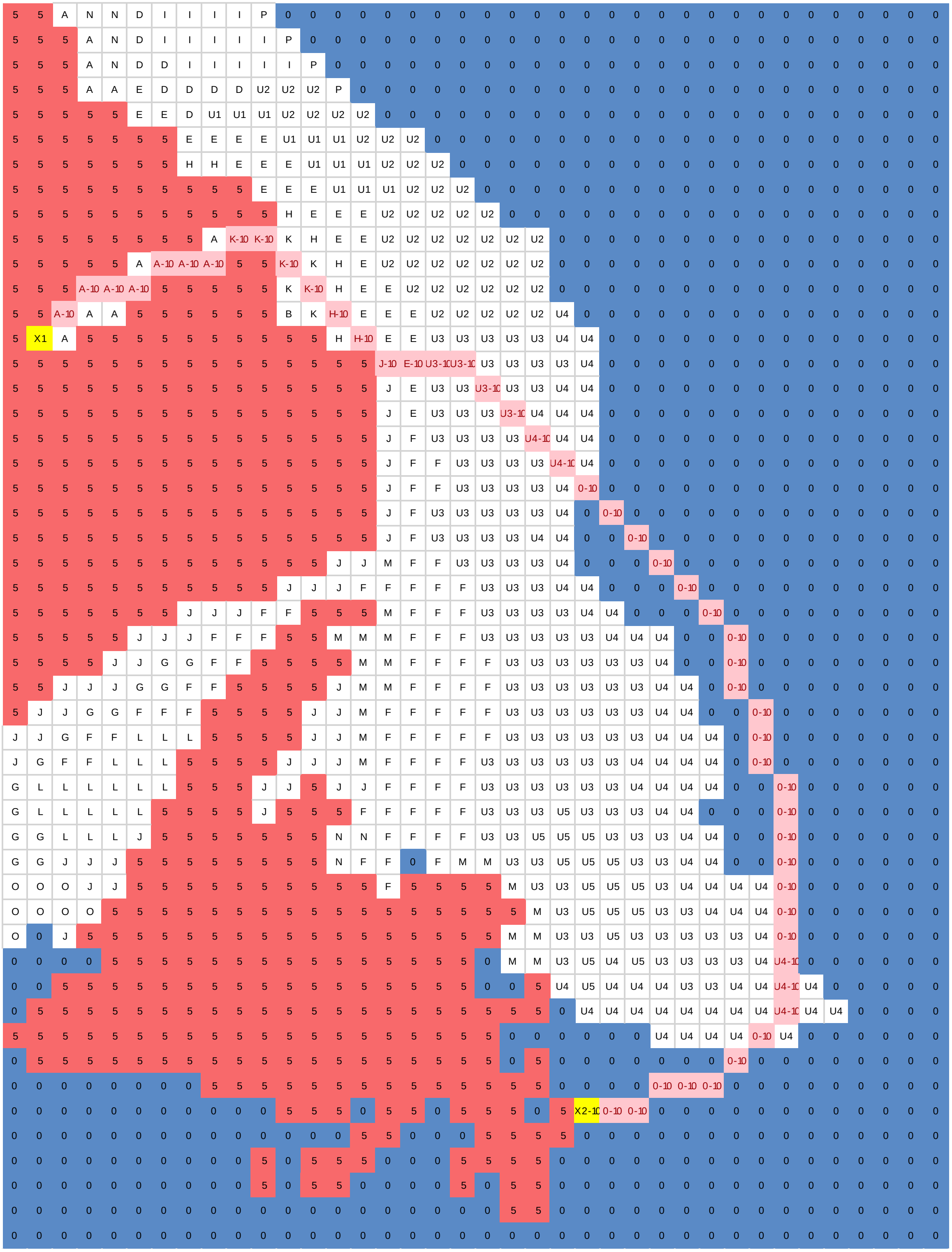

Figure 19. Route of PC5 with ice drifting

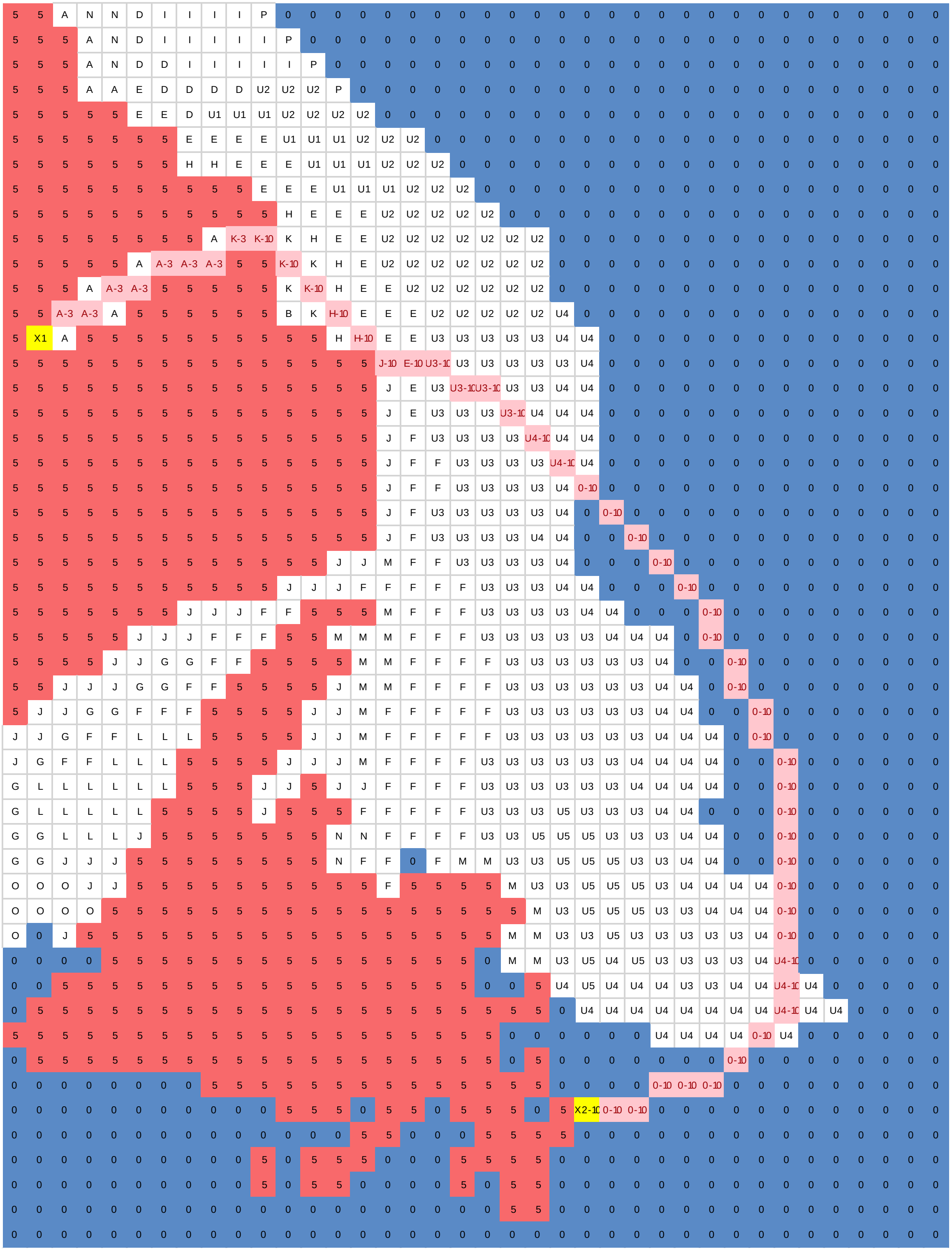

Figure 20. Route of PC7 with ice drifting

Figure 17, Figure 18, Figure 19, and Figure 20 are the results of the routes of PC5 and PC7 with weights (0.1, 1, 4). The model only finds routes for PC5 and PC7 but cannot find a permissible route for IC. The reason for this is that the ice in the area surrounding Goose Bay is too severe for this ice class. With 10 per tenth of ice concentration of medium first-year ice (regime A), the RIO calculated of this area is -40. It is clearly not permissible for this ice class to operate in these conditions.

As can be seen from the results, the routes of PC5 and PC7 are similar. The strategy of the voyages involves moving away from ice instead of sailing along the coast through the navigable ice (RIO > -10). The only difference is the navigation in regime A near Goose Bay. PC5 can travel at full speed, while PC7 needs to reduce to 3 knots due to safety concerns. The RIOs of other regimes of both ships are positive, so there is no other constraint. As a result, their behaviours for the rest of the voyage should be the same.

### 4. Route from Goose Bay to St. John's with temporal change of the environment

The fourth investigation is an expansion of the third trial, with consideration of the temporal change of the environment. The ice estimation in Newfoundland waters updated by CIS is on a daily basis during the winter. According to the previous estimation, it takes nearly three days for the ship to voyage from Goose Bay to St. John's. Hence, this trial uses the historical data from the Canadian Ice Service Ice Archive to figure out the optimal route for the ship on a multiple-day trip. Three daily ice charts are used, starting from March 11, 2020 (Figure 3, Figure 21, and Figure 22). This evaluation reuses all configurations from trial 3 with the added consideration of ice drifting. The new RIO of the last two days is calculated in Table 15 and Table 16. The route suggestions for each day are shown in Figure 23, Figure 24, and Figure 25. Note that the route planning for each day should use the corresponding ice chart. The same ice egg codes might have different values in different ice charts. Figure 26 is the concatenation of all sub-routes overlaying on the ice chart of the first day.

Table 15. New RIO on day 2

| **Regime** | **0** | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** | **J** | **K** | **L** | **M** | **N** | **O** | **U1** | **U2** | **U3** | **U4** | **U5** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **PC5** | 30 | 10 | 20 | 23 | 15 | 25 | 26 | 30 | 18 | 30 | 28 | 30 | 29 | 30 | 30 | 26 | 24 | 21 | 27 | 29 | 30 |

Table 16. New RIO on day 3

| **Regime** | **0** | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** | **J** | **K** | **L** | **M** | **N** | **O** | **P** | **Q** | **R** | **U1** | **U2** | **U3** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **PC5** | 30 | 10 | 20 | 23 | 15 | 25 | 26 | 30 | 27 | 30 | 21 | 30 | 30 | 30 | 30 | 30 | 30 | 29 | 30 | 21 | 27 | 26 |

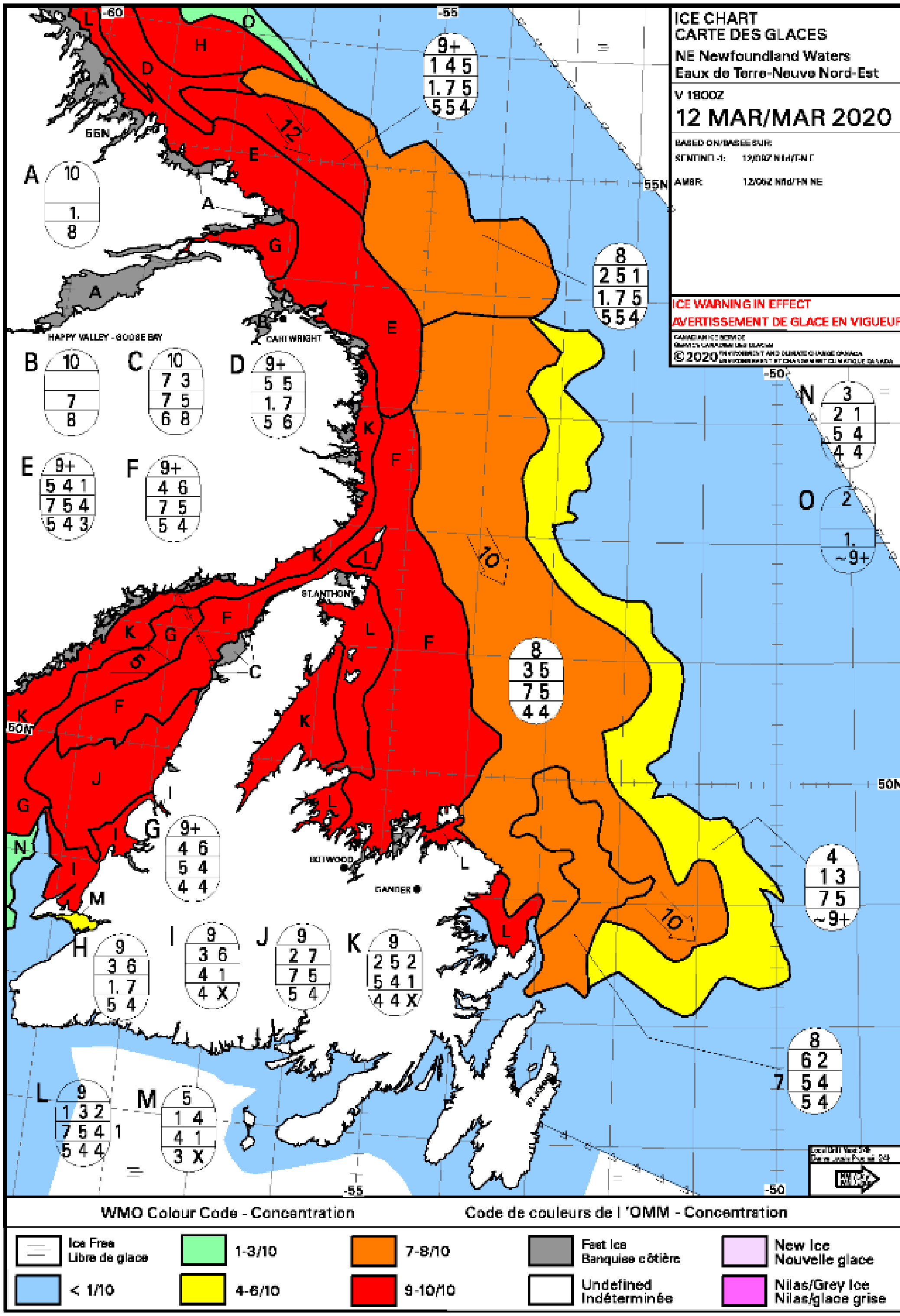


Figure 21. The ice chart of Newfoundland waters on Mar 12, 2020 (Ice Archive, 2019)

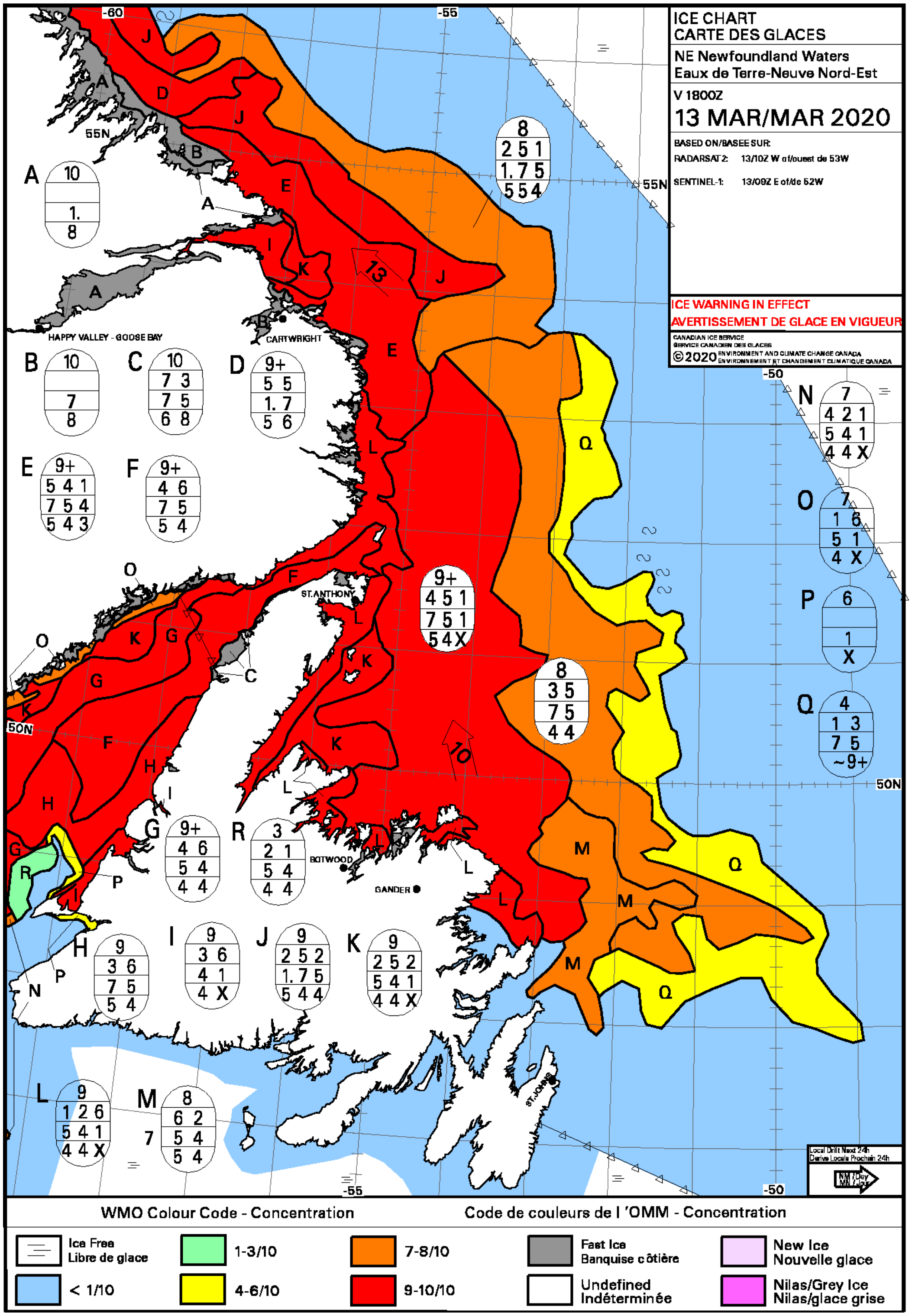


Figure 22. The ice chart of Newfoundland waters on Mar 13, 2020 (Ice Archive, 2019)

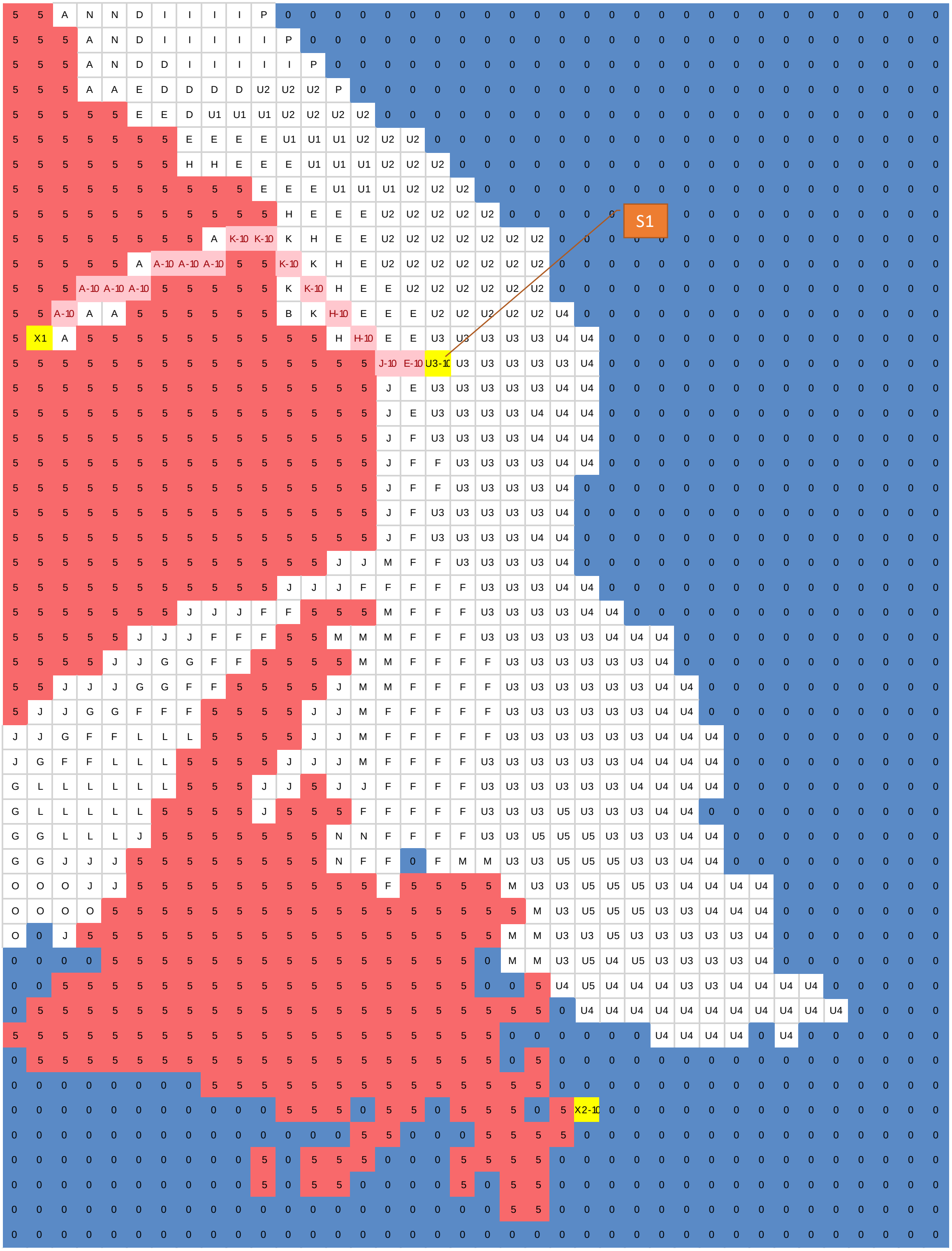


Figure 23. Route for March 11, 2020 – day 1

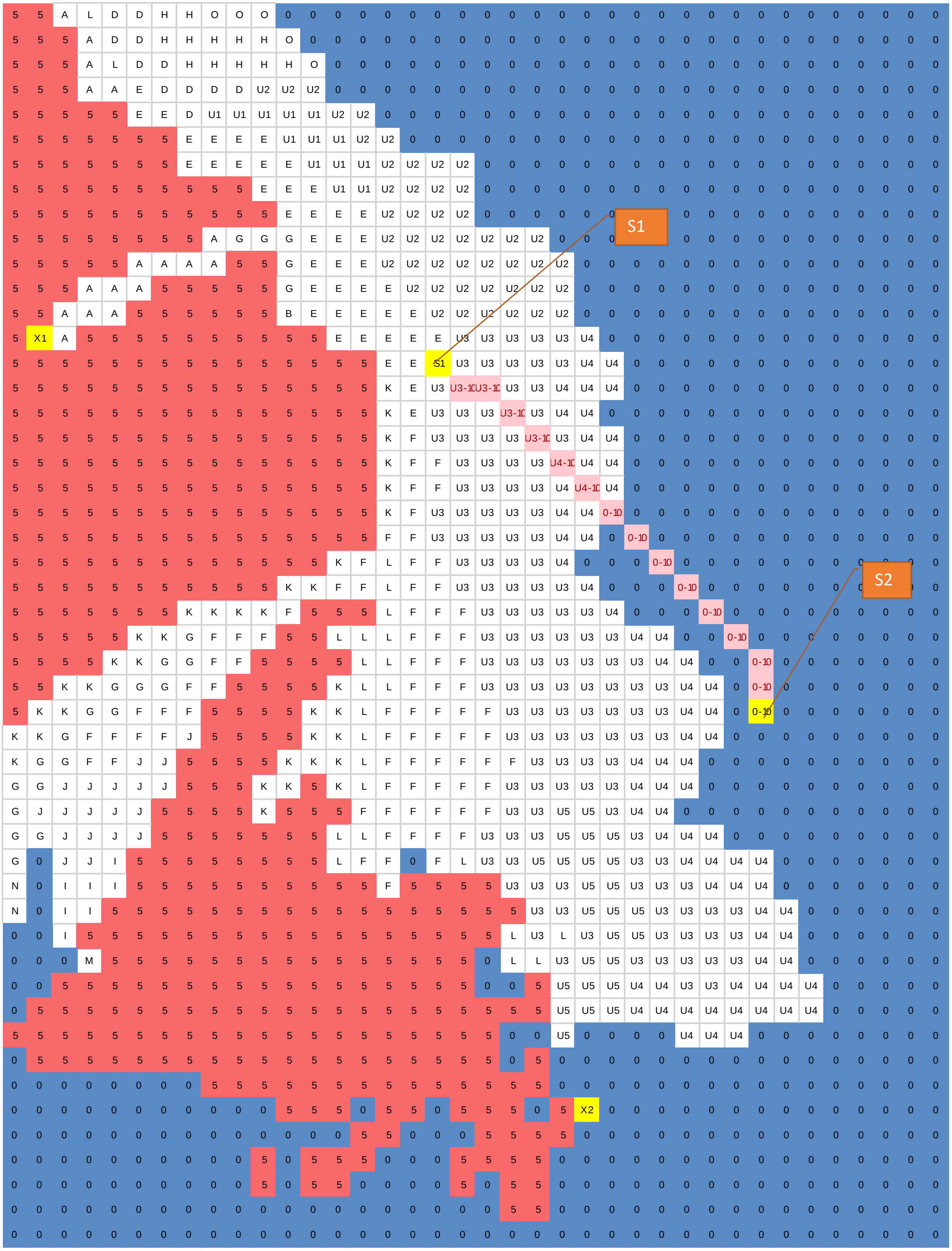


Figure 24. Route for March 12, 2020 – day 2

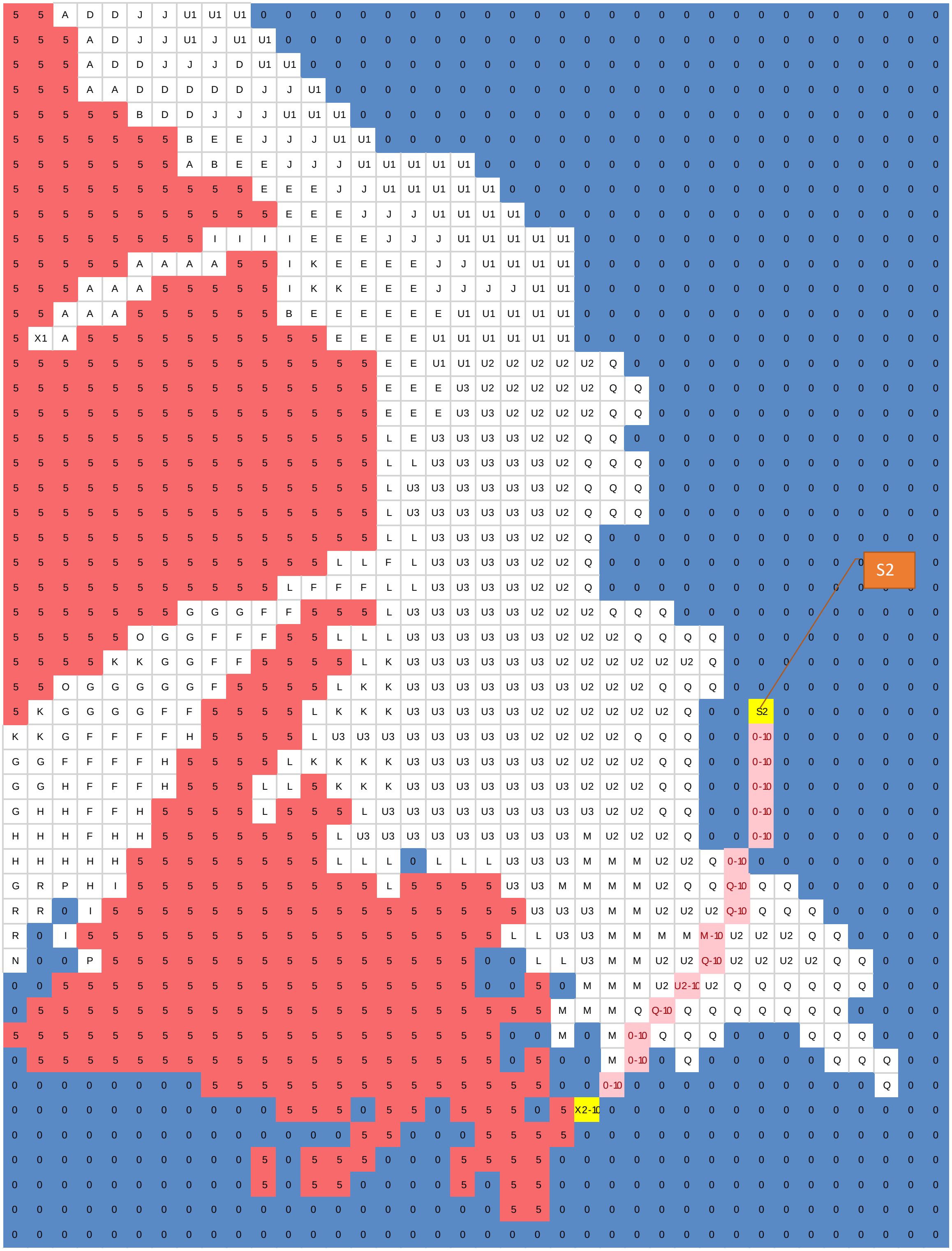


Figure 25. Route for March 13, 2020 – day 3

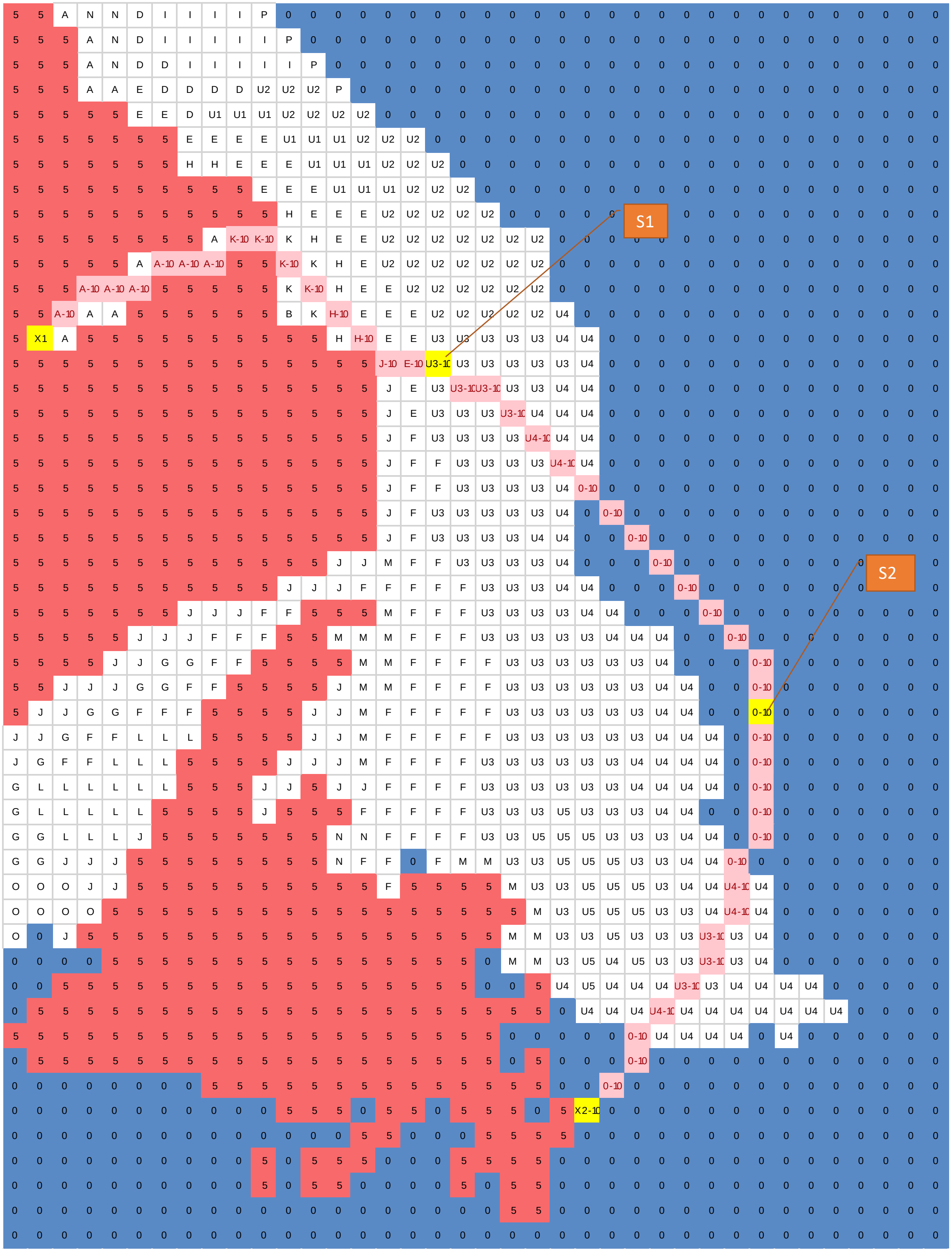


Figure 26. Full 3-day trip from Goose Bay to St. John's

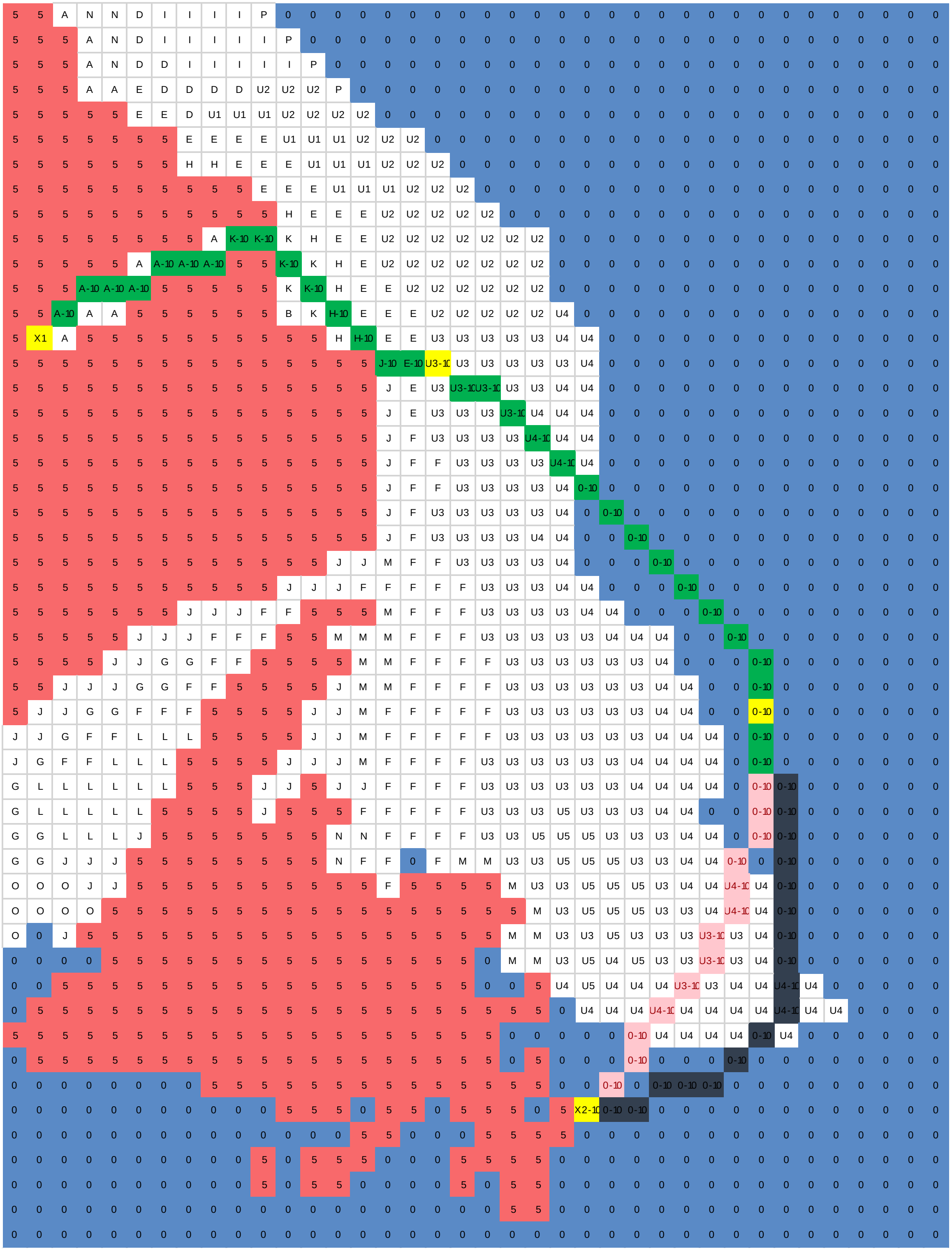

Figure 27. Comparison to old route (green: common area, red: new route, grey: old route)

As a result, the ship PC5 takes approximately three days to voyage on the water. On day 1, the ship travels from X1 to S1 (Figure 23), then from S1 to S2 on the second day (Figure 24) before heading to X2 on the final day (Figure 25). This result is not the same as that of trial 3 (Figure 27). The new ice charts cause the difference on day three. The new data reflect the new status of all regimes.

## 5. Validation

A validation was conducted with experienced captains. The purpose of this step is to compare the results of this research to the routes taken by experts. The specific scenario is considered by experienced captains who have lots of knowledge and skills in navigating in ice-covered water. Two experts were contacted individually to make sure the results were independent. They were asked to plot a route they would take for a merchant ship (PC5) transporting cargo from Goose Bay to St. John's, as in Figure 3. The basic information of the ship was provided to the captains in Table 17. Specific constraints and objectives such as optimizing distance, time, and fuel consumption were not mentioned in the request to these captains.

Table 17. Ship particulars (Frederking, 2003)

| **Beam** | **16 m** |
|---|---|
| Waterline length | 75 m |
| Draft | 6.5 m |
| Displacement | 5000 tonnes |
| Design speed | 10 knots |
| Ice class | PC5 |

The routes generated by the two captains are displayed in Figure 28 and Figure 29. The first captain shows the route starting from St. John's (point A) to the Hamilton Inlet (Regime K), then navigates the ship west to the open water as soon as possible. The speed is set around 6 knots for the ice section. Once away from the ice, he takes the route that keeps at least 10 miles away from the ice edge to St. John's with full speed at 10 knots. The second expert chooses the same route from Goose Bay to Hamilton Inlet but sets the speed at 3 knots. He mentions that the 120 cm ice is difficult in steerage. He speeds up the ship to 8-10 knots in regime K and slows down to 7-8 knots in H and 6-7 knots in E areas. He would go southeast on 8/10 coverage with 7-8 knots before arriving in the open water. Full speed is his choice for the rest of the voyage. The captain takes a risk to cross the thin ice to reduce the distance travelled in the final stage.

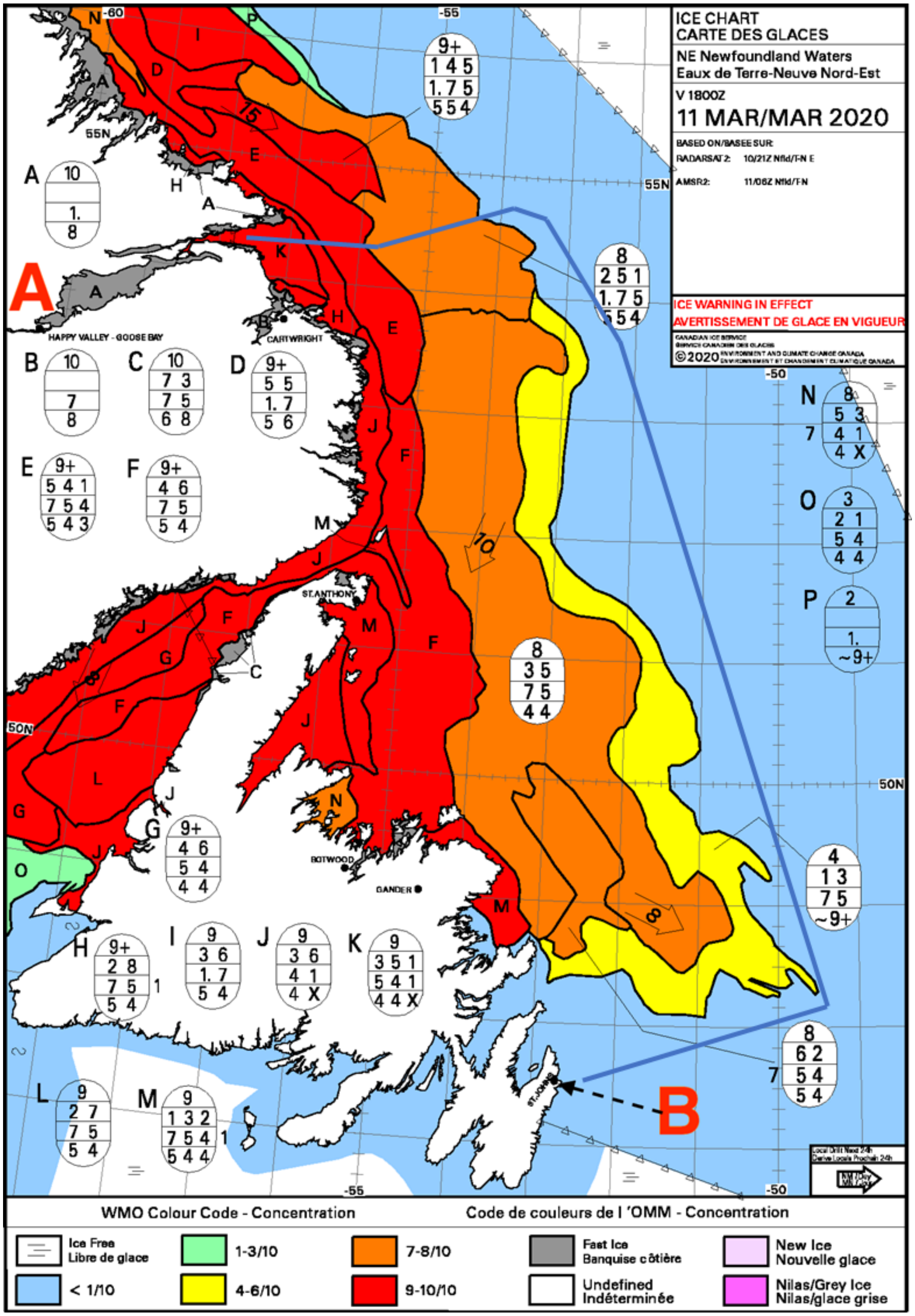


Figure 28. Ice chart of Newfoundland on March 11, 2020 and route selected by Expert 1

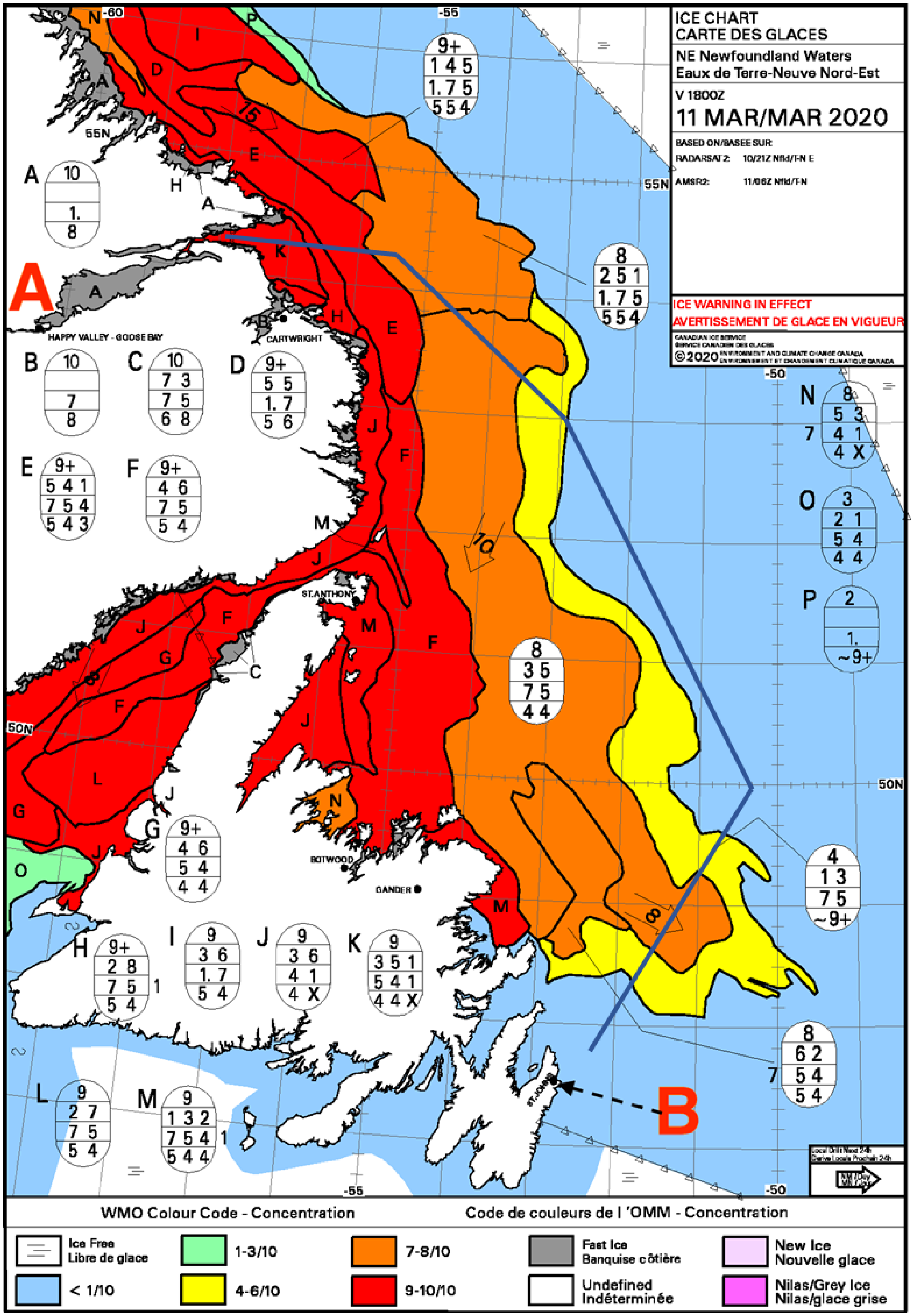


Figure 29. Ice chart of Newfoundland on March 11, 2020 and route selected by Expert 2

# Chapter 5: Discussion

This chapter discusses the optimality in this multiple objective decision-making problem, comparing reinforcement learning and graph search. The impact of non-determinism and ice classes on route planning is also presented. Last but not least, the practical application is analyzed.

## 1. Optimality

The ultimate goal of this project is to identify the optimal routes for ships in a specific environment. The solution must optimize competing objectives, such as sailing distance, voyage time, and fuel consumption. The ideal optimality is that the best solution has minimal distance, time, and fuel at the same time. Let us consider the problem with only two objectives, time and fuel consumption when the ship travels in open water given the same distance. Intuitively, if the ship wants to arrive at the target in a short time, it needs to operate at the highest speed. This decision leads to high fuel consumption to boost the engine power. Conversely, if the ship wants to use the least fuel amount, it will use a lower speed, so the ship will take a longer voyage time in this scenario. The ideal case where the ship can attain the lowest voyage time and the lowest fuel consumption is unrealistic. Therefore, there does not exist the ideal optimality for the multiple competing objective problems.

The next concern is finding good enough results as the solution. In other words, an alternative definition of optimality should be proposed for the decision-making process. Pareto dominance is an excellent way to evaluate and compare two solutions for a multi-objective problem. Nevertheless, when the number of competing objectives increases, the quantity of optimal solutions by Pareto grows significantly (Tozer et al., 2017). This research uses a scalarization method to transform a vector of all objectives into a scalar signal. The best solution is now the one having the highest score. Different weights in the scalarized process might generate different optimal results, but all of them are relatively optimal based on the justification of the system designers.

In trial 3, a grid search of settings is run as Table 14. The setting (0, 0, 1) shows the model focuses only on fuel consumption. The result will ensure that the route has the smallest fuel amount. Similarly, (0, 1, 0) and (1, 0, 0) help us know the boundary of time and distance travelled, respectively. If the model chooses these weights, only one objective is optimized, and the trip has to pay a high cost for other objectives. For example, let us consider the case (0, 0, 1) with the ship PC5. The result shows the best fuel consumption at 69 tonnes, but the voyage takes four times longer in terms of time than the results having the best time, at 262 hours vs. 62 hours. The ship also travels about 1.5 times farther than the shortest route (917 NM vs. 620 NM). Hence, the other combinations of three objectives are investigated to search for a reasonable solution. Based on Table 14, the result of (0.1, 1, 4) is nominated as a potential route for both PC5 and PC7 commuting from Goose Bay to St. John's. This route is 73 hours, 11 hours more than the best time, while the fuel consumption is 79 tonnes, 10 tonnes more than the best fuel value. The route is approximately 112 NM longer compared to the shortest route.

In summary, the optimality of a multi-objective problem is a relative term. This conceptual framework is a means to find a suitable solution based on a predefined relationship of all objectives. The advantage of this model is that it is tunable. Reverse engineering can be executed based on the real data from the shipping industry to mimic the expected experience.

## 2. Reinforcement learning and graph-based methods

Trial 1 and trial 2 show the comparison of the reinforcement method and Dijkstra's algorithm. The final solutions are the same. The similarity of these results of the two methods is as expected. It is because the reward function of reinforcement learning and the cost function of Dijkstra's algorithm is constructed in the same way. They find the same optimal route in the first trial where the probability of ice drifting $p = 0$. In trial 2, Dijkstra's algorithm cannot solve the problem when $0 < p < 1$ as it only works in the deterministic case. The graph-based method can only handle the problem when $p = 1$, which means the ice drifting always happens. On the other hand, the output of the Q-learning method is a deterministic policy, although the environment and transition of the agent might be stochastic. In this situation, the Q-values will converge to the policy when $p = 1$. Therefore, these methods should generate the similar results.

In terms of the time metric, the Q-learning takes much more time than the graph search to solve the problem, as in Table 10 and Table 11. This result can be explained by that the reinforcement learning spends thousands of episodes on learning about a known environment, whereas the graph-based method does not take time for the learning process. It chooses the best action greedily and only runs one episode from the start point to the endpoint. In reality, graph search is one of the best methods to solve the pathfinding problem when the environment is fully observed. The reinforcement learning is too general in this situation. The learning through the interaction between agent and environment is unnecessary if the environment is already known.

In summary, Dijkstra's algorithm outperforms reinforcement learning to solve a pathfinding problem with a known environment. Both methods have the same results, but Dijkstra's algorithm solves the problem much faster.

### 3. Deterministic and non-deterministic environment

Trial 1 investigates the deterministic case, and trial 2 considers the non-deterministic one. Compared to trial 1, trial 2 is more realistic. By interacting with the environment and the existence of the uncertainty, the agent can learn this lesson and decide to choose the safer route. However, coming up with a good result costs runtime. It takes more time for agents to learn about this uncertainty using reinforcement learning compared to the deterministic case. The long run time might be acceptable if it does not exceed a limit. Planning a day route within an hour is not the worst result. More discussion about the application will be mentioned in the next section.

### 4. Temporal dynamics of the environment

Trial 4 investigates how temporal changes in the environment influence the route. The result of trial 4 is more optimal and realistic than the former results in trial 3. Nevertheless, there is an assumption that the 3-day ice prediction is already known in advance. This trial is implemented based on the historical data of the ice chart. In reality, the future ice chart is unknown. The ideal case is to have a good forecast model for planning the route better at the beginning of the trip. By this, the planned route is optimized along the way. Otherwise, the planning process needs to update periodically according to the new data release of the ice chart.

### 5. Validation with two captains

Both captains indicated that they are familiar with ice classes in the Arctic Shipping Pollution Prevention Regulations (ASPPR) (Transport Canada, 2009) and AIRSS regulations for the planning process because they have used this information for years. According to Transport Canada (2009), PC5 in the Polar Class of the International Maritime Organization is equivalent to a class between Type A and CAC 4 in the ASPPR system. This conversion is necessary to calculate the Ice Numeral by the AIRSS regulation.

In terms of AIRSS, the guidelines are similar to POLARIS. In the scope of this research, the specific comparison of these two regulations is not performed. However, with specific ice charts like Figure 3, Figure 12, and Figure 13, the PC5 is not restricted by any constraints of AIRSS and POLARIS. In other words, the ship will have the same route when either of these regulations is applied.

The routes suggested by the experts are compared to the results from trial 3. In terms of direction, the first captain is more risk-averse. He prefers open water to the ice by navigating the ship to ice-free areas as soon as he can and following the ice edge to sail to his destination. The second captain takes more risk, heading to some thin ice regimes. The result of the agent in this research looks more similar to that of the second expert.

When it comes to the speed, both of the experts decide to operate at slow speed in thick ice, mid-range speed when ice is less severe, and full speed in open water. Meanwhile, the agent opts to run full speed for the whole voyage, no matter the ice. These captains explain their selection by the engine power and the uncertainty associated with operation in ice. Firstly, the engine power of a specific ship limits the possible speed, especially when it operates in thick ice areas. Secondly, the visibility factor in ice regimes causes difficulty in navigation. The captain needs to reduce the speed, for example, in foggy weather and operation at night. Finally, the second captain warns that the ship should exercise the standard mariner caution since the Labrador coast might have old ice or bergs anywhere, but they are usually not shown in the ice chart. The agent in this research does not consider these factors. The decision to run at 10 knots is because the operation has the best score based on the framework defined by the rewards. According to Equation (6), the engine power required to run at 10 knots in a 1.2m ice regime is greater than 25 MW. This is a lot of power, but the assumption is that the ship can have unlimited power. The time of the voyage is also shortened with the full-speed operation.

Furthermore, the first expert mentions that he would keep at least 10 miles from the ice edge when in open waters. This decision is analogous to the model because the ships also stay one cell away (12 nautical miles) from the high risk in as mentioned earlier in trials 2 and 3.

In summary, there are some similarities and differences between the results of the research and the realistic routes as the analysis above. Generally, the direction chosen by the agent is comparable with the routes of experts. The speed selection might be adjusted when all other factors are considered.

# Chapter 6: Conclusion and Future Work

This research focuses on the pathfinding problem with multiple competing objectives for ships in ice. The objectives include distance travelled, voyage time, and fuel consumption, together with the POLARIS constraints. A conceptual framework is proposed as an end-to-end solution where Dijkstra's algorithm and Q-learning are used. The optimality depends on a linear scalarization process. The framework is evaluated in an idealized environment to find routes with both deterministic and non-deterministic situations. The results show that the graph-based method outweighs the Q-learning in both cases due to the fully-observed environment. Although Dijkstra's algorithm cannot be applied for the probabilistic environment, it can find similarly optimal results as Q-learning does. In both scenarios, the running time of Dijkstra's method is much less than that of Q-learning to solve the same problems. Besides, the framework also tackles the real pathfinding problem from Goose Bay to St. John's in Northeast Newfoundland using the ice chart from CIS. The complexity develops from a static environment to a daily changing one. A reference check is implemented by two experienced captains to validate the result. In conclusion, this conceptual framework is an expandable and tunable tool for multi-objective route selection for the ship in ice. The more sophisticated the model is, the more practical the results are. Therefore, this research has potential to become a realistic application.

In the future, there are some areas to improve this research. Firstly, the fuel consumption calculation is simple in this model. Future work should investigate a better method to estimate fuel consumption. Secondly, the research uses a linear scalarization technique to simplify the objective vector. Subsequent work might look for a better way to compare vector to vector, such as non-linear scalarization. Thirdly, though this project includes POLARIS as the safety constraint, it is not enough for safety concerns. Navigating in ice should comply with other guidance to reduce the structural damage for ships. Fourthly, increasing uncertainty in the pathfinding model is an essential step to make the route more practical. These factors could be the visibility, the existence of old ice, as discussed earlier. Fifthly, the next research should solve the multi-criteria pathfinding in ice under uncertainty by leveraging graph-based algorithms. Finally, the incorporation with experienced seafarers is essential to gain the domain knowledge and expertise to make the model more realistic.

# References


Canadian Ice Service. (2016, March 7). *Interpreting ice charts*. Canada.ca. **https://www.canada.ca/en/environment-climate-change/services/ice-forecasts-observations/publications/interpreting-charts/chapter-1.html.**

Censor, Y. (1977). Pareto optimality in multiobjective problems. *Applied Mathematics and Optimization, 4*(1), 41-59. **https://doi.org/10.1007/bf01442131**

Choi, M., Chung, H., Yamaguchi, H., & De Silva, L. W. A. (2013). Application of genetic algorithm to ship route optimization in ice navigation. *Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions (POAC)*, Espoo. **https://www.poac.com/Papers/2013/pdf/POAC13_167.pdf**

Choi, M., Chung, H., Yamaguchi, H., & Nagakawa, K. (2015). Arctic sea route path planning based on an uncertain ice prediction model. *Cold Regions Science and Technology, 109*, 61-69. **https://doi.org/10.1016/j.coldregions.2014.10.001**

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms*. MIT press.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik, 1*(1), 269-271. **https://doi.org/10.1007/bf01386390**

Frederking, R. (2003). A model for ship routing in ice. *Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions (POAC),* Trondheim. **https://www.poac.com/PapersOnline.html**

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley.

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics, 4*(2), 100-107. **https://doi.org/10.1109/tssc.1968.300136**

Ice Archive. (n.d.). [Archived ice charts from Canadian Ice Service]. Retrieved December 01, 2020, from **https://iceweb1.cis.ec.gc.ca/Archive/page1.xhtml**

International Maritime Organization (2016). *MSC.1/Circular.1519 - Guidance on methodologies for assessing operational capabilities and limitations in ice*. **https://www.imorules.com/MSCCIRC_1519.html**

Keinonen, A., Browne, R. P., Revill, C. R., & Reynolds, A., (1996). *Icebreaker Characteristics Synthesis, Icebreaker Performance Models, Seakeeping, Icebreaker Escort* (*Vol. 1).* Transportation Development Centre.

Kotovirta, V., Jalonen, R., Axell, L., Riska, K., & Berglund, R. (2009). A system for route optimization in ice-covered waters. *Cold Regions Science and Technology, 55*(1), 52-62. **https://doi.org/10.1016/j.coldregions.2008.07.003**

Lehtola, V., Montewka, J., Goerlandt, F., Guinness, R., & Lensu, M. (2019). Finding safe and efficient shipping routes in ice-covered waters: a framework and a model. *Cold regions science and technology, 165*, 102795. **https://doi.org/10.1016/j.coldregions.2019.102795**

Liu, X., Sattar, S., & Li, S. (2016). Towards an automatic ice navigation support system in the Arctic Sea. *ISPRS International Journal of Geo-Information, 5*(3), 36. **https://doi.org/10.3390/ijgi5030036**

Piehl, H., Milaković, A. S., & Ehlers, S. (2017). A finite element method-based potential theory approach for optimal ice routing. *Journal of Offshore Mechanics and Arctic Engineering, 139*(6). **https://doi.org/10.1115/1.4037141**

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Tozer, B., Mazzuchi, T., & Sarkani, S. (2017). Many-objective stochastic path finding using reinforcement learning. *Expert Systems with Applications, 72,* 371-382. **https://doi.org/10.1016/j.eswa.2016.10.045**

Tran, T. T., Browne, T., Peters, D. & Veitch, B. (2020, November). A reinforcement learning approach to route selection for ice-class vessels. *The 29th Annual Newfoundland Electrical and Computer Engineering Conference (NECEC)*. IEEE.

Transport Canada. (2009). *IACS Unified Requirements for Polar Class Ships - Application in Canadian Arctic Waters.* Ship Safety Bulletins, Bulletin No.: 04/2009. **https://tc.canada.ca/en/marine-transportation/marine-safety/ship-safety-bulletins/bulletin-no-04-2009**

Transport Canada. (2018, March 03). *TP 12259E - Arctic Ice Regime Shipping System (AIRSS) Standard.* **https://tc.canada.ca/en/marine-transportation/marine-safety/tp-12259e-arctic-ice-regime-shipping-system-airss-standard**

Van Moffaert, K., & Nowé, A. (2014). Multi-objective reinforcement learning using sets of pareto dominating policies. *The Journal of Machine Learning Research, 15*(1), 3483-3512.

Van Moffaert, K., Drugan, M. M., & Nowé, A. (2013, April). Scalarized multi-objective reinforcement learning: Novel design techniques. *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)* (pp. 191-199). IEEE.