2022

Agent-based Modelling and Big Data: Applications for Maritime Traffic Analysis

Marc SIRIZZOTTI
siri2950@mylaurier.ca

Agent-based Modelling and Big Data: Applications for Maritime Traffic Analysis

by

Marc Sirizzotti

MSc Geography, Wilfrid Laurier University, 2022

THESIS/DISSERTATION

Submitted to the Department/Faculty of Geography

in partial fulfilment of the requirements for

Master of Science in Geography

Wilfrid Laurier University

Abstract

Agent based modeling (ABM) is a powerful tool for examining complex systems in many scientific applications, including maritime transport systems. Growing demands for freight transport and increased industry emphasis on reducing environmental impacts have heightened the focus on vessel and port efficiency. This research aimed to create a maritime route planning model to simulate vessel movement in all waterways. The goal of the ship routing model developed in this research was to develop a simulation tool capable of reproducing real world shipping routes useful for navigation planning, with emphasis on port scheduling and potential application for further use and exploration. A modified breadth-first search algorithm was implemented as a NetLogo ABM in this research. With increasing volumes of ship location monitoring data, new approaches are now possible for examining performance-based metrics and to improve simulations with more precise verification and analysis. A Satellite Automatic Identification System dataset with over 500,000 vessel logs travelling across the Pacific Ocean and into the Port of Metro Vancouver was used as the focal area for model development and validation in this study. Automatic identification system (AIS) is the global standard for maritime navigation and traffic management, and data derived from AIS messages can be used for calibrating simulation model scenarios. In this analysis, the results examined how changes in simulation parameters alter route choice behaviour and how effective large AIS datasets are for validating and calibrating model results. Using large AIS datasets, model results can be quantified to examine how closely they resemble real-time vessels in the same region. Heatmaps provide a data visualization tool that effectively uses large data sets and calculates how closely model results resemble AIS data from the same region. In the case of PMV, the

Maritime Ship Routing Model (MSRM) was able to replicate path likeness with a high level of accuracy, generating realistic navigation paths between the many islands on the eastern side of southern Vancouver Island, B.C., a busy marine traffic region and sensitive ecological area. This research highlights the use of ABM as a powerful, user-friendly tool for developing maritime shipping models useful for port scheduling and route analysis. The results of this study emphasize the use of large data sets that are applicable, clean, and reliable as a crucial source for validating and calibrating the MSRM.

Acknowledgements

I would like to start by thanking Dr. Colin Robertson for his support and patience. For taking on my master's project at the later stages of progress when he did not have to. I am thankful to have finished this chapter of my life and although there have been many difficulties, I am grateful for the lessons it has taught me. I would also like to thank the graduate department for providing me with extensions needed to complete my research and GSTS and Exact Earth for providing me with the data. I would like to thank all the professors and department staff who have helped me get to this point, along with committee member Dr. Steven Roberts for aiding in my research. I would also like to thank my family and friends for the support that was needed during difficult times. Without all the support, I would not have been able to accomplish this research.

Table of Contents

List of Figures

1. Introduction

1.1. Big data GIS

The ability to acquire, share and process large quantities of data has fundamentally changed the way researchers study the world. The term big data has been coined to represent increasingly large datasets that have outstripped our ability to manage and preform analysis, requiring new techniques and tools that take advantage of this new paradigm(Haug, 2016). For geospatial research, growing sources and variety of data via satellites, location sensors, online activity, and other ubiquitous sensors has significantly increased the volume and variety of data available to with location information (Graham & Shelton, 2013). These new data sources expand research capabilities by utilizing modern technological advancements, allowing researchers to effectively use big data to study new problems or tackled existing research problems from new perspectives (Mirovic, Milicevic, & Obradovic, 2018). Often, big data is described using the three V's: (1) volume— the amount of data that can be collected and stored; (2) velocity— the speed at which data can be captured; and (3) variety— encompassing both structured (organized and stored in tables and relations) and unstructured (text, imagery) data (Miller, Goodchild, 2015). Further descriptions of big data expand on the 3 V's, touching on several key features including, value— increased amounts of data does not provide value without methods of extracting useful information and providing outcomes. Veracity— the accuracy of the data, including the quality, integrity, and credibility. Collaboration can be required to collect big data and merging sources may be difficult; thus, attention to accuracy across sources is crucial. Variability— the changing nature of data. In geography this can include the differences in scale, temporal and spatial distribution and attributes (Kitchin, 2013).

New software and sensors are collecting large amounts of data, much of which contain a geographic context and provide a valuable resource for researchers and industries (Graham & Shelton, 2013). Domain-specific applications of geographical analysis and modelling such as the logistics and transportation industry can benefit greatly from new research techniques that help to facilitate the adoption of new data sources. It has been noted that the steep decrease in storage costs, new wireless technology development and the low-cost widespread usage of sensors, both personal and industrial, have provided researchers with the ability to gain valuable insight into the transportation industry (Torre-Bastida et al., 2018). Location data collected from global positioning systems (GPS) and more specifically automatic identification systems (AIS), have become standard in the maritime sector.

The growing amount of data regarding maritime navigation and performance brings many possibilities including; detecting and predicting vessel activities, enhancing safety, detecting anomalous activities and to support critical decision-making (Vouros, Doulkeridis, Santipantakis, & Vlachou, 2018). AIS data have been used to enrich many facets of maritime research examples of which are discussed further in section 1.5 Agent Based Modelling and Big Data GIS: Maritime Examples.

1.1.1. Big Data GIS: Challenges

The use of big data has created many hurdles for researchers to overcome, including having to deal with messy unorganized datasets, ignoring spurious patterns, limited access, and effectively using big data to build data driven models (H. J. Miller & Goodchild, 2014).

All data sources require processing to acquire and store, convert to new formats, and identify missing and/or anomalous values. These common tasks become more complicated with big data where there is much more data to handle (Blazquez-Soriano & Domenech, 2017). Erroneous data are also of concern as sensors and satellites can malfunction, failing to collect data or incorrectly record. Other possibilities include mistakes in manual entry. In the transportation sector, there is also the prospect of deliberate errors reporting misleading data which may be in favor of operators for economic incentives or illegal activity (Mirovic et al., 2018). Solutions include using more frequent automated data collection however, this does not guarantee improvement in all contexts. Alternatively, extensive validity checks can be applied to minimize errors. This includes replacing incorrect data where possible, anomaly detection and filling incomplete data (Mirovic et al., 2018). Methods for automated and manual cleaning strategies may not always be suitable. Certain approaches may remove useful information by cleaning noise and data outliers that may be pertinent to the dataset (Vlahogianni, 2015). Finally, even with clean data flawed inferences can be drawn from spurious correlations obtained from the analysis of big data (Calude & Longo 2017).

Access to big data sources may be difficult to acquire for many independent researchers, academic institutes, and small businesses. Many big data sources are collected by large private businesses and governments with restricted, limited, or expensive access requiring researchers to negotiate access (Kitchin, 2013). Furthermore, when access is gained there are often several securities and ethical challenges required to work with such data sets. When large data sources happen to be shared more readily, they are often void of sensitive or important data.

The increase in shipping demand and the globalization of the world economy has increased the number of vessels at sea and consequently the amount of AIS data (Norris, 2006).

However, too much data can be a concern as it may be harder to interpret, process, store, and data collection can place strain on existing infrastructure (Sui, Goodchild, & Elwood, 2012).

1.2. Data Visualization and Heatmaps

Data visualization tools are important for graphical representation and effective interpretation of datasets. The goal of data visualization is to explore data by discovering relationships, patterns and differences that may be impossible to identify with statistical procedures or viewing spreadsheets. Secondly, data visualization tools can provide important context when interpreting data that aid users' understanding and inspire further ideas and new hypotheses. These tools support the presentation of data that helps tell a story and provide a visually appealing method for discovering and conveying patterns or relationships in data.

The strengths of data visualization tools are well suited to big data presentation and analysis. Evan Sinar (2020) compares the benefits of data visualization tools with the main facets of big data: volume, velocity, and variety. Volume- increasing amount of information can be too large to interpret effectively without visual representation. Velocity- visualization tools provide a data structure that is easily and quickly updated as the speed of data retrieval is important for users understanding the new, incoming data. Variety- the ability to show trending and time-series data is a strength of data visualization tools and provides the ability to visually align and integrate data from a variety of scales and sources.

Heatmaps are graphical representations of the spatial variation in density of a spatial process (Słomska-Przech, Panecki, & Pokojski, 2021). Heatmap analysis can be applied to all types of geographic vector data (point, line, polygon), however is most frequently used to explore spatial variation in spatial point process data (Kulyk & Sossa, 2018). Heatmaps are the visualization of data points using a color gradient to represent the weight or influence of a certain point. Areas of influence surrounding each point can overlap, further impacting the values and color gradient (Netek, Pour, & Slezakova, 2018). With reliable data cleanup, heatmaps can be used to explore and identify single instances or clusters of important data in large datasets. Without such visualization techniques, effectively using large datasets may be challenging(Anderson, 2009). Transportation problems are particularly suited to heatmap analysis using GIS platforms (Słomska-Przech et al., 2021), which can provide a tool for examining changes in spatial patterns of vessel traffic over time (Netek et al., 2018).

1.3. Needs of the Maritime and Transportation domain

A maritime port has many safety, efficiency, and environmental concerns that are heavily regulated and constantly monitored. The Maritime port authority enforces certain rules and regulations to ensure the smooth running of all traffic and activity in a port and collects ongoing data to improve functions and inform policy for the future.

The regulations and policies include restrictions on speed/ size/ weight, anchorage, fuel consumption and efficiency, right of way, scheduling and routing, bridge usage and scheduling, interactions with local wildlife particularly endangered species, unsafe cargo management, waste management, traffic control/ collision avoidance/ safety, preventing nefarious activity, documentation, and immigration (*Port Metro Reference Guide*, 2016). The optimization of port efficiency and enforced regulation is imperative to the financial success of the port economy, the wellbeing and safety of the human population living in the area as well as those on the ships, the survival of the wildlife, and limiting damage to the natural ecosystems.

Data analysis is a necessary method of making use of collected records of activity in the port to improve policy, regulation, and enforcement. There are several ways in which data can be used and analysed to improve functionality and policy in a Port, several examples are covered in section 1.5 Agent Based Modelling and Big Data GIS: Maritime Examples. Agent based modeling (ABM) has the capacity to simulate ship pathways and produce improved path plotting to save time, money, and increase fuel efficiency (Helmreich & Keller, 2011).

Simulation modelling methods are particularly useful for studying the maritime and transportation domain. They demonstrate a state of movement, examine interactions amongst agents, and calculate various time parameters to determine influences on the flow of transportation systems, making it possible to study and improve operations (Guo & Hu, 1994). Many applications of simulation modelling are used in the transportation domain. For example, Goerlandt and Kujala (2011) examine ship collision probability using AIS data to obtain realistic input data for traffic simulation using a collision detection algorithm. Their findings provide detailed information on the circumstances of ship encounters including; location, encounter angle, time, size and speed of vessels etc. The results are valuable for consequence analysis which can be incorporated with probability analysis directly obtained from the model to provide an idea of risk level in different ship encounters (Goerlandt & Kujala, 2011). Often transportation systems are comprised of large-scale networks with complex interactions. They are also driven in part by unpredictable behaviour. These circumstances lend themselves well to the implementation of Agent Based Modelling (ABM)(Kagho, Balac, & Axhausen, 2020). ABM has the capacity to simulate ship pathways and produce improved path plotting to save time, money, and increase fuel efficiency (Helmreich & Keller, 2011). Detailed description of ABM is covered in 1.4 and subsequent sections.

Big data analytics can be used to explore many challenges in maritime traffic analysis. Freight markets as a field of academic study have increased in relation to the commodity price boom of the early 2000's. They are an integral part to the success of globalization and the global economy. Specific shipping routes and cargo flows can be studied to capture economic trends and processes movement. This has implications for market analysis forecasting freight rates and ship capacity.

There are many environmental and sustainability related concerns in the maritime system including waste management, fuel usage, and conservation and protection of endangered species. Reduction of fuel consumption is one of the areas of research that has received the most attention in recent years. The ability to design ships and routes more efficiently has huge implications for reducing greenhouse gas emissions from freight transportation in North America and water quality/ habitat conservation in the area, which as been estimated at 4% of global CO^2 emissions (Bialystocki & Konovessis, 2016).

The commercial shipping Industry has a negative impact on natural habitats and the safety of endangered sea life. Marine mammal conservation is a priority for the Canadian Government and various environmental groups exist such as the Institute of Cetacean Research, The American Cetacean Society, Whale and Dolphin Conservation, The Ocean Alliance, and the Ocean Conservation Research. The government of Canada has invested in scientific research under "The Whales Initiative." The Whales initiative seeks to increase knowledge of locations, movement, and population of Whales in Canada. One of the projects the Canadian Government is currently investing $9.1 Million dollars in is The Whale Detection and Avoidance Initiative, which is part of Canada's Oceans Protection Plan("Report to Canadians: Investing in our coasts through the Oceans Protection Plan," 2021). The Whale Detection and Avoidance Initiative is funding various projects that are developing new technologies to detect whale locations in real-time. Having better information on where whales are located could help mariners avoid colliding and thus injuring or killing them. The current projects are utilizing underwater microphones on fixed and mobile platforms, and infra-red cameras. One possible avenue for expanding on the currently collected data would be to gather the information of the whales' locations and simulate potential interactions with vessel traffic. Simulating likely routes for whales in relation to most frequent routes for ships would highlight areas where collision is most likely to occur.

The Government of Canada is also investing $26.6 million into the "Marine Environmental Quality" (MEQ) initiative ("Report to Canadians: Investing in our coasts through the Oceans Protection Plan," 2021) through the Oceans Protection Plan. In collaboration with outside partners, Department of Fisheries and Oceans Canada researchers are conducting research exploring the impact of shipping-related noise on marine mammals. These projects are focused on endangered St. Lawrence Estuary Beluga, North Atlantic Right Whale, and Southern Resident Killer Whale. The data collected on noise and its effects on marine life could similarly be used to create ABMs and explore how various changes made to shipping routes may affect the noise level on marine life. A further $3 million dollars is currently being invested by the government of Canada into projects that are working to understand risk factors for marine mammals and inform policy and protection. Better understanding and promoting conservation of sensitive marine ecosystems is a research area of increasing significance in Canada and around the world. Given the capabilities of ABM to explore ‘what-if’ scenario planning, there may be significant application of these tools in the marine/environment domain.

1.4. What is Agent Based Modelling?

Agent Based Modelling is a computational tool used for modeling and simulating complex systems by representing the behaviors of agents and the processes by which they interact. The objects, also known as agents, follow specific sets of rules that can be observed and manipulated. ABM is well suited for examining heterogenous and dynamically changing processes, assessing the interactions of agents and overall impact on complex systems (J. Huang et al., 2022). This is particularly useful for situations where agents are in motion and behaviour can be predicted, such as navigation or traffic analysis (Davidsson, Henesey, Ramstedt, Törnquist, & Wernstedt, 2005). Simulations can run many scenarios by altering parameters to examine how small changes can appear at the macro level. Analyzing model results can provide a better understanding of a problem and determine further courses of action.

One benefit to ABMs over other forms of modelling such as mathematical modelling, is the ability to model more abstract non-mathematical forms such as verbal models. Verbal Models can help describe the relationships informally as rules or principles in natural language. This can make it possible to model system behaviours which are not known prior to model development(Scheutz & Mayer, 2016).

1.4.1. What is an agent?

Agent Based Models are dependent upon the incorporation of autonomous entities called agents. Agents are individuals, groups of individuals or organizations represented in the model, attributed with decision making entities and governed by a set of rules. Each agent is programmed to be defined with specific properties as well as its relationship to other agents. All actions and interactions carried out by agents are directed by the behaviours initially attributed to them. Stochasticity can be applied to agent behaviour when randomness of behaviour patterns is necessary. Using the set of rules and objectives assigned to each agent, the agent makes decisions on its behaviour based on situational assessments. Some actions that an agent may take are producing, consuming, selling, moving, recording, spawning, living and dying, etc.

ABM allows for extreme diversity and heterogeneity to be programmed into agent characteristics and agent interactions, as well as in dynamics, adaptation, and feedbacks (Yu, 2002). Traditional statistical models are unable to accommodate and represent this level of variety and detail (Barbati, Bruno, & Genovese, 2012). Thus, research questions which involve a substantial amount of heterogeneity and diversity in agent interactions and scope are well suited to ABMS. Complex behaviour patterns, emergent behaviour and information about the dynamics of real-world systems can be produced with simple ABMs (Kagho et al., 2020). Emergent behaviour is that which arises from the interactions of discrete parts of a system and cannot be easily determined or extrapolated from the individual behaviour of agents(Helbing & Balietti, 2015).

1.4.2. History of Cellular Automata and The Game of Life

Cellular Automata (CA) is a model theory that consists of an array of cells that represent a discrete spatial confine (typically two-dimensional) where each cell processes inputs on characteristics resulting in various "states"(Crooks, 2017). This spatially distributed process is governed by a set of rules that determine the state of cells based on that of their neighbouring automata. CAs can appear simple however, they have the potential to complete various computations and simulate real-world processes (Crooks, 2017). The History of cellular automata begins with John von Neumann who proposed the idea as a model of self-reproducing organisms (Sarkar, 2000). CAs are a simpler version of agent-based models, however they provide an effective introduction to dynamic simulation modelling in general and how complexity can emerge from a small set of static rules. ABMs are multi-agent systems with emphasis on platform development and do not have a discrete spatial extent or a fixed number of cells (Clarke, 2014). CAs typically follow a smaller set of rules that are always governed by neighbours, providing fewer applications.

A classic example of a CA model that also exhibits the properties of ABM is "The Game of Life." This simulation is a straightforward model that produces complex results with interesting potential for experimentation. "The Game of Life" is a simple two-dimensional grid. Each cell on the grid can be either alive or dead. One set of cells are randomly assigned their state (living or dead) at the commencement of each game. Each game iteration generates a new assigned state for the cells at random. The Agents in this model are the cells, and their basic behaviour options are that they are either alive or dead.

The rules governing how the agents react during the game are as follows,

"1. Any live cell with fewer than two neighbours alive dies.

2. Any live cell with two or three neighbours' alive lives on to the next generation.

3.Any live cell with more than three neighbours alive dies.

4. Any dead cell with exactly three neighbours alive becomes a living cell"

(Adamatzky, 2010)

ABM and CA models, even those with a simple set of rules such as “The Game of Life” produce observable patterns over time that can be extrapolated on, studied, and experimented with. This ability to simulate activity and interactions throughout a system over time to discern emergent patterns is one of the key strengths of ABMs. These complex interactions and macro effects of the system cannot be easily determined from analyzing the rules of the game (Mi Yu, 2015). Refer to figure 1.1 for a visualization of patterns that can emerge from Conway’s Game of Life.

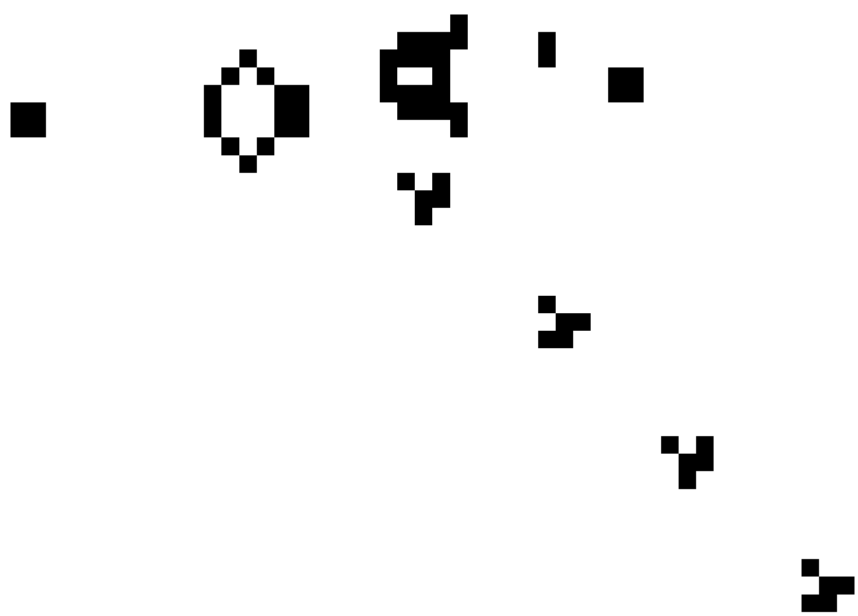


Figure 1.1**:** Introduction to agent-based models and cellular automata. Highlighting repetitive complex patterns that can emerge after many iterations of Conway’s Game of Life. (Stevens, 2019)

1.4.3. Agent Based Modelling Applications

Agent based modelling is actively being applied in many fields of academic study and commercial research. A few examples are in the field of medical research, the dynamics and spread of diseases have been explored using agent-based models. (e.g., Eubank et al., 2004).

Energy flows through a power grid were examined by Pacific Gas and Electric using an agent-based model (Bonabeau, 2003a). The effects of hiring strategy on corporate culture were examined using agent-based models at Hewlett-Packard (Bonabeau, 2003). The effect of changes to decimalization on the stock market was explored using agent-based modeling by NASDAQ (Bonabeau, 2003b; Darley and Outkin, 2007). In ecology, agent-based models have been used to simulate the migration and evolution of salmon populations (Railsback and Harvey, 2002). Emergency response planning has used agent-based modelling to simulate and improve processes around wildfire training. This includes implications for incident command and community outreach (Guerin and Carrera, 2010). The dynamics and flow of vehicle and pedestrian traffic were explored using agent-based models (Helbing and Balietti, 2011). Agent-based Models were used in the drug development process by Eli Lilly (Bonabeau, 2003a).

Transportation systems are well suited for agent-based studies because they are geographically distributed in a dynamic changing environment (Bo & Cheng, 2010). All disciplines of transportation research have examples of well-known ABM platforms. Some examples include Transportation Analysis and Simulation System (TRANSIMS), Multi-Agent Transport Simulation Toolkit (MATSim), Sacramento Activity-Based Travel Demand Simulation Model (SACSIM), Simulator of Activities, Greenhouse Emissions, Networks, and Travel (SimAGENT), Open Activity-Mobility Simulator (OpenAMOS),and Integrated Land Use, Transportation, Environment (ILUTE) (Hong Zheng, 2013). Most examples follow a similar structural design. All modelling platforms contain agents that represent an individual traveller (human or vehicular) with attributes and characteristics that govern their behaviour. Activity plans adhere to system demands and can be revised to meet the needs of various constraints (spatial, temporal, etc.).

1.5. Agent Based Modelling and Big Data GIS: Maritime Examples

The ability to validate and calibrate existing simulation models with AIS and other geographic data has created many possibilities. Tesfatsion et al. (2006) describe three methods of validating computational models (Tesfatsion & Judd, 2006).

1. Descriptive output validation - analyzing computational data by comparing results to available real-life data. This is the most intuitive of all methods and is fundamental in accurate calibration of computational models.
2. Predictive output validation - this method uses computationally generated data to predict data that has not been captured yet. This can be a problem as model calibration and validation may not be possible at the time of conception. However, many models need to be predictive as their implementation is useful for future forecasts and preparation.
3. Input validation - By analyzing actual data, researchers can introduce the correct parameters to the model before operation. This can be valuable for creating realistic model environments, but also the most difficult to apply when accompanied with ABM (Bianchi, Cirillo, Gallegati, & Vagliasindi, 2007).

The majority of research uses descriptive output validation as a means for analysis. Navigation and ship performance efficiency often uses past data to conduct research for supply chain, environmental and other maritime queries. Perera et al. (2015) examines various vessels and environmental factors (wind speed, direction, engine power, shaft speed and fuel consumption) to optimize trim configuration. Trim configuration has a direct impact on energy efficiency and with the available data, statistical analysis can compare minute changes. By analyzing past trends in weather and ship movement, optimizing route and more specifically trim position can increase fuel efficiency and improve logistics (Perera, Mo, & Kristjánsson, 2015).

Examples are less frequent but do exist using the other validation methods described by Tefatsion et al. 2006. For example, The Maritime Research Institute Netherlands (MARIN) used an existing simulation technology "Dolphin" and incorporated the ability to read AIS data. Using Dolphin and real data, researchers attempted to replicate traffic scenarios to test decision making processes of autonomous ships. With a focus on risk assessment and collision avoidance, the findings are capable of contributing to the greater dynamic safety assessment model (Brake, Iperen, Looije, & Koldenhof, 2015).

A recent example by Kanamoto et al. (2021) using predictive output validation, estimated the global trade flow pattern of dry bulk cargo using AIS data. Combining different data sources, the authors were able to forecast vessel type demand and trade volumes (of certain commodities). Due to the variety of sources and amount of data needed, research of this nature would not be possible without big data analytics and AIS data(Kanamoto, Murong, Nakashima, & Shibasaki, 2020).

Ali Akbar Safaei et al (2019) used AIS data from ships to create models that explored fuel consumption predictions. The study used data from the Noon Report (NR) and AIS in its study. The data characterizing four Very Large Crude Carriers (VLCC), was used to create a prediction model. The fuel consumption rate was determined by considering several factors, ship displacement, ballast water and bunker, average daily sailing speed, trim and sea conditions (wind, wave and current) and cargo. The formula proposed:

$$Fc = 1393 + 116.6Vv + 0.001Ds + 4.94Wv \quad (1)$$

Where Fc is fuel consumption (tons/day), Vv is ship velocity (k), Ds is ship displacement (metric tons), and Wv is Wave height (Beaufort scale). This formula predicted the fuel consumption of the vessels in various conditions and these predictions agreed with the recorded fuel consumption data. The researchers' recommendations were that in future studies, nonlinear regression methods should be applied to increase accuracy of predictions (Safaei, Ghassemi, & Ghiasi, 2019).

Nefarious activities, such as the smuggling of illegal cargo such as drugs, human trafficking, and piracy is an age-old problem that international organizations and national security organizations are concerned with tracking, preventing, and ending. Piracy activity in particular costs the shipping industry billions of dollars in losses. In 2008 pirate activity led to a loss of 16 billion dollars (Jakob, Vaněk, & Pěchouček, 2011; O. H. Ondrej Vanek, Michal Jakob, Michal Pechoucek, 2011).

A recent study utilizing Agent-Based traffic management techniques by a team of engineers at the Czech Technical University in Prague tackled the problem of Pirate attacks on vessels in the Gulf of Aden (Michal Jakob, 2012). The team combined agent-based modelling methods and simulation of maritime traffic and novel route planning and scheduling algorithms. An ABM called Agent-C: Agent-based System for Securing Maritime Transit was developed to anticipate the movements of modern maritime pirates in relation to international shipping routes. The Agent-C model focuses on three types of behaviours, patrolling, shipping, and piracy.

The shipping aspect of the program tracked the routes that vessels use to transport cargo between multiple locations. This aspect is relatively straightforward and is programmed to choose pre-existing routes that are most efficient in both time and cost (fuel consumption), however, factoring in security of the passage significantly increases complexity.

Piracy behaviour preys on the reliable and predictable nature of shipping routes, and therefore choosing routes that only factor in efficiency of time and fuel create route paths that are more likely to be targeted by pirate ships. Pirate vessels discover, approach, and attack other small to medium size vessels. They then hijack the vessels and escort it to pirate bases. Pirate operations have varying levels of technology at their disposal, from basic roaming the area for victims to employing radars, AIS data monitoring, and mothership gangs that work in unison (Michal Jakob, 2012). Agent-C included patrolling behaviour in their modelling simulation, currently utilized by various security forces. Security vessels patrol pirate infested waters to discourage and halt piracy behaviours. Patrolling is one of the most effective deterrents of piracy behaviours.

The results of programming shipping, piracy, and patrolling into a single ABM created an incredibly complex system (Jakob et al., 2011; Michal Jakob, 2012; M. J. Ondrej Vanek, Michal Pechoucek, 2013; O. H. Ondrej Vanek, Michal Jakob, Michal Pechoucek, 2011; O. H. Ondrej Vanek, Michal Pechoucek, 2014; Vanek, 2013; Vaněk, Jakob, Hrstka, & Pěchouček, 2013). The systems created were so complex that arriving at a clear solution was determined to be infeasible due to the large number of vessels involved and the complexity of their relationships with their scheduling and route paths.

Three recommendations were made based on the results of the simulations. Ideally all three recommendations would be implemented together for the best possible results to improve maritime security. Employing transit routes and patrolling patterns that can minimize the likelihood of attack. The ABM was able to simulate these possibilities by maximizing utility and minimizing the importance of risk and time. The simulations that employed stochasticity into the routes experienced a twofold drop in attack rate.

The Agent-C model ended up modelling significant complexity between ships, pirates, patrols, and their environment leading to a very computationally demanding model. Finding the optimum patrolling policy was not straightforward and designing the optimal routing policy to ensure security was not feasible. Agent based techniques deployed in the Agent-C experiments demonstrated potential for improving maritime security, although also revealed the challenges and limits of overly complex simulations in a real-world setting.

1.6. Agent Based Modelling Challenges

The application of ABM provides a useful tool for researchers but is not without its own set of challenges. Anticipating whether a model is suited to a particular theory or application, in advance, can be difficult and at times impossible. In ABM the outcomes of interactions are inherently unpredictable. Identifying how programming directly affects emergent model behaviour is challenging due to the complex and open-ended nature (Railsback, Lytinen, & Jackson, 2006).

Replicating scientific research is pertinent in all fields of study. The more instances where a model can be replicated indicates a greater ability to inform our understanding of the phenomenon. By studying independent situations and repeating analysis with different software and programming, researchers can ensure repeatability. However, ABM may be hard to replicate using different software and programming languages. Due to the nature of ABMs, issues with transparency can make it hard to explain and replicate results (Kagho et al., 2020). Detailed information regarding commands and primitives are often not available to consumers (Hong Zheng, 2013). Without knowing how certain functions operate at a detailed level, it may be hard to reimplement coding from one language to another.

Many ABMs aim to simulate a real-world problem where plentiful data sources can be used to validate and calibrate results. Using a good, representative dataset is needed to provide useful and accurate information for model production and analysis. Validation identifies to what extent the model represents the system being studied. The validity of a model should not be treated as a binary event but rather involve a goodness of fit test to examine how well it answers the research question (Crooks, Castle, & Batty, 2008). Calibration involves adjusting key model parameters to reflect the behaviour of a real system more closely (M. J. Ondrej Vanek, Michal Pechoucek, 2013). This is related to validation as adjusting model parameters relies on identifying the goodness of fit. Validation and calibration are crucial in the development of any effective model but are only possible where data already exists and can be gathered.

The goal of agent-based modeling is to create simulations that hopefully produce outcomes useful to researchers. Developing model simulation environments from scratch was once necessary to study agent-based modeling applications. This is effective as it allows developers access to all minute details of computation. However, it can be difficult to generalize models to other situations. The development of agent-based modelling toolkits such as Repast, Swarm and Netlogo provide an excellent environment for implementing geo-spatial ABMs. They help researchers focus on the construction of models without having to also build the fundamental tools and building blocks required to produce computer simulation (Railsback et al., 2006).

GIS is a pertinent tool in analyzing input and output of a geo-spatial nature. They are not developed with dynamic modelling such as ABM as their primary concern (Maguire, 2005), however, linking GIS software and ABMs is important to facilitate analysis of simulations and take advantage of the strengths of both software types. Modern toolkits provide options to output data and communicate directly to GIS software platforms. For example, Netlogo has a GIS extension, that contains commands providing external functions necessary for combining GIS analysis.

1.7. Path Finding Algorithms

Path finding algorithms exist as a method to search nodes in a weighted graph and find the shortest path. This is accomplished by using a starting node and selecting new nodes until a desired destination is reached. In graph theory, the shortest path problem is defined as finding a path between two vertices such that the sums of the weights of path edges is minimized (Mathew, 2015). The 2 objectives of path finding algorithms are; successfully finding a path between two nodes and, choosing the route with the smallest cost(Delling, Sanders, Schultes, & Wagner, 2009).

Basic algorithms such as breadth-first (BFS) and depth-first search (DFS) will calculate and explore all possible solutions. BFS involves an iterative loop over a queue of vertices computing the cost from the given source vertex to all other reachable vertices in a layered fashion (Holdsworth, 1999). The algorithm will visit, check and or update all un-visited nodes in a tree-like structure. Figure 1.2 is an example of BFS traversing through a series of nodes layer by layer.

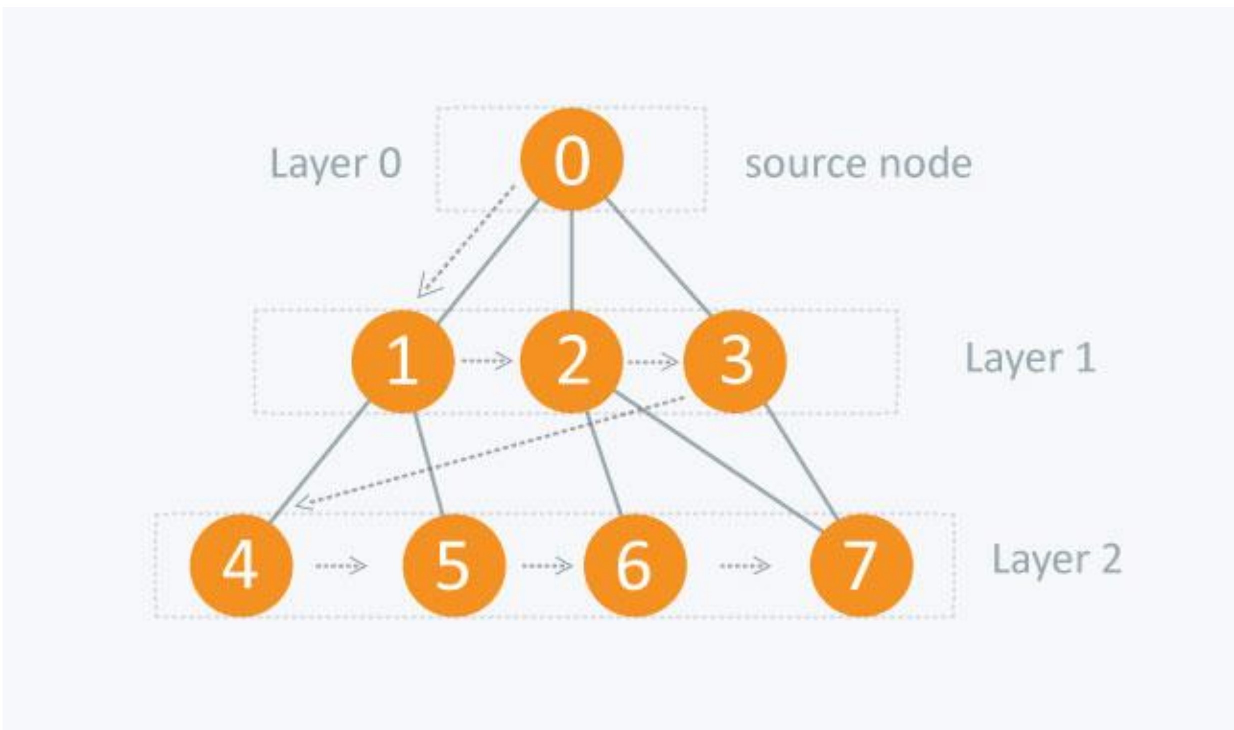


Figure 1.2: BFS search algorithm queue order (Garg, 2022).

A graph G is typically represented as G = (V, E) where V are vertices and E are edges. In BFS, the edges of G connected to a source node are traversed computing the distance to all reachable vertices. This process repeats iteratively until the final node has been located, when the search will stop and the path with the shortest cost is determined. The BFS approach is often used as a building block for other algorithms as it has low computational intensity but can result in low overall performance when compared to other algorithms (Beamer, Asanovic, & Patterson, 2012). Optimized algorithms such as Djikstra's and A* are more efficient as they ignore previously examined nodes and will only choose nodes with the lowest value, eliminating less optimal paths once the entire graph has been calculated (Rachmawati & Gustin, 2020). Figure 1.3 provides graphical representation of both algorithms in the presence of an obstacle.

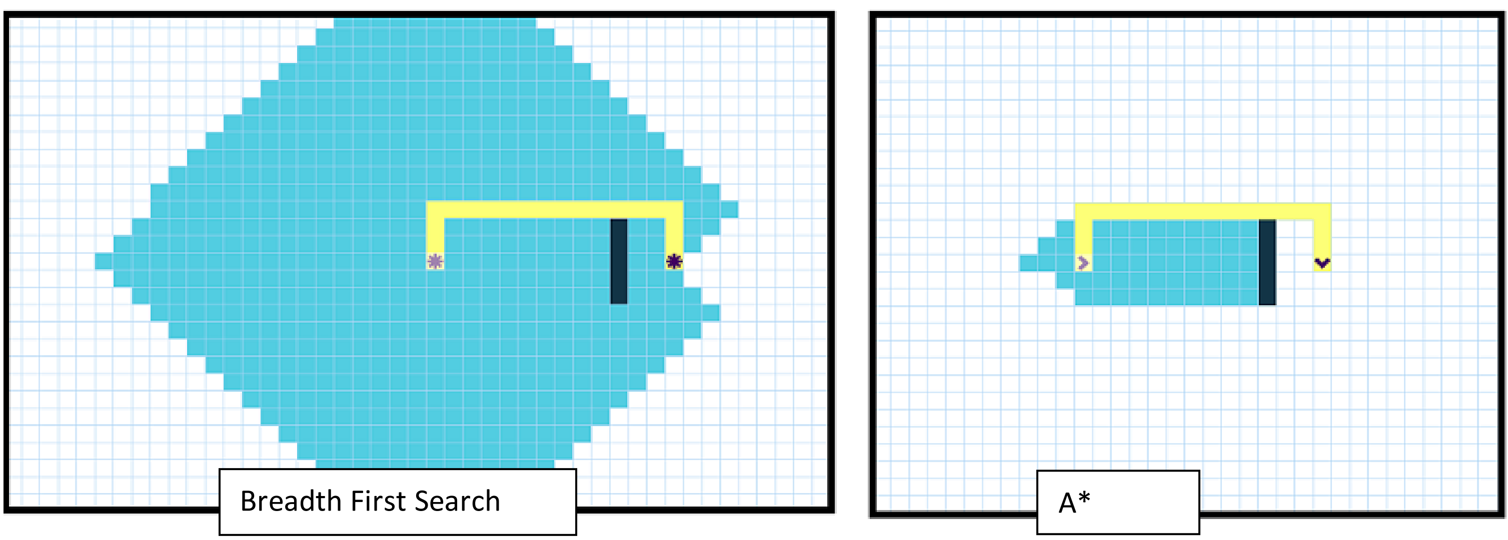


Figure 1.3: Graphical representation of BFS and A* search algorithm in the presence of an obstacle. Yellow line represents the chosen path, light blue is the search area and dark blue is an obstacle (Mihailescu, 2019).

A* path finding algorithm is a variant of Djikstra's algorithm that includes heuristics. Heuristics is defined as a rule (or set of rules) intended to increase the probability of solving some problem. In the case of A* the heuristic determines the approximate distance to the destination as the minimum possible distance between the node and the end (Mathew, 2015). This enables the algorithm to eliminate longer paths once it determines the initial path. Figure 1.4 gives a visual representation using a network of nodes to show how the A* algorithm works.

7	6	5	6	7	8	9	10	11		19	20	21	22
6	5	4	5	6	7	8	9	10		18	19	20	21
5	4	3	4	5	6	7	8	9		17	18	19	20
4	3	2	3	4	5	6	7	8		16	17	18	19
3	2	1	2	3	4	5	6	7		15	16	17	18
2	1	0	1	2	3	4	5	6		14	15	16	17
3	2	1	2	3	4	5	6	7		13	14	15	16
4	3	2	3	4	5	6	7	8		12	13	14	15
5	4	3	4	5	6	7	8	9	10	11	12	13	14
6	5	4	5	6	7	8	9	10	11	12	13	14	15

Figure 1.4: A* path finding example. Where node 0 (green) is the starting node, and node 19 (blue) is the destination. Red nodes represent the path selected according to the A* algorithm. Gray nodes represent an obstacle.(Swift, 2020)

A* is calculated using the formula f = g + h, where f is the total cost of each node, g is the distance from the current node and start node, and h is the heuristic, in this case is the estimated Euclidean distance from the current node to the end node. There are three main forms of distance heuristics: Euclidean, diagonal shortcut, and Manhattan. Manhattan distance is the standard heuristic used with a square grid, as it only allows for movement in cardinal directions (4 directions). Diagonal shortcut can be applied when the grid allows for movement in 8 directions. Euclidean distance allows for movement at any angle regardless of grid directions - providing straight line distance between points. Because of this, Euclidean distance is always shorter than Manhattan or diagonal distance. Euclidean distance is calculated as follows:

$$d_{euc} = \sqrt{(x_{path} - x_{goal})^2 - (y_{path} - y_{goal})^2} \quad (2)$$

where (x_{path}, y_{path}) are the coordinates of the current node and (x_{goal}, y_{goal}) are the coordinates of the goal. In this heuristic the distance between 2 points is a straight line. This is a simple approach to calculating heuristics but more computationally expensive (Leigh, Louis, & Miles, 2007) than other heuristics.

1.8. Research Objectives

The objectives of this research are as follows.

1. Explore potential path finding algorithms to use in a maritime ship routing scenario.
2. Discover how ABM can be used to simulate maritime ship routing by developing a modified path finding algorithm to suit a maritime ship routing scenario.
3. Evaluate different scenarios of model parameters and data cleaning/preparation to determine what modifications provide the best representation of ship behaviours observed in AIS data.
4. Identify future model functions and research directions to address key issues for maritime routing in the context of global-scale S-AIS based data.

2. Methods

2.1. Model Objectives

The overall aim of modelling was to recreate vessel movement at port or in hard to navigate waterways by combining spatially explicit AIS and bathymetric data in an ABM environment. By modifying the BFS path-finding algorithm to suit a maritime example, simulation results attempt to replicate *in situ* route selection. Modification will identify boundaries where ground and water deep enough for cargo vessels meet based on bathymetry data.

Various path finding algorithms were tested in early model development. Informed algorithms such as A* provided more efficient path finding, determining the shortest path using single start and end points faster than uninformed search algorithms like BFS. However, they do not provide the specific needs for algorithm modification and maritime simulation. To replicate realistic shipping routes the algorithm should not prioritize finding the optimal path, but instead modify the selected path to resemble the route taken by shipping vessels in the same region. Furthermore, A* will not calculate shortest path values for all nodes in the waterway, potentially failing to identify ground boundaries necessary for algorithm modification. The MSRM allows for ships to visit many anchors before a final destination, replicating the various anchors maritime shipping vessels may encounter before reaching port. With BFS each anchor can act as the new start point as the cell it occupies contains shortest path values for all other anchors and ports. Without a search algorithm that exhausts all cells in the network, path finding values for anchors at any location would not be possible. This also allows hundreds of ships to be added at any location in the network without having to calculate shortest path values as each start cell already obtains BFS values for all anchors and ports.

GIS techniques via NetLogo extensions and QGIS are utilized to analyze and visualize simulation results and compare them to the AIS dataset of the same region. Using large datasets as a basis for verification of model results is an effective and relatively new phenomenon (Kavak, Padilla, Lynch, & Diallo, 2018). AIS data set with hundreds of thousands of signals collected in 2012 are used to verify model results. The modular architecture of the model aims to be useful for implementing additional simulation criteria such as ship-to-ship interaction, port/anchorage procedures, and fuel consumption analysis.

2.2. Data and Software

2.2.1. What is AIS data?

Automatic Identification System (AIS) is a form of communication used by maritime vessels to broadcast and receive information regarding ship identity and location. Sharing information between ships and land-based receivers that govern maritime traffic is pertinent for recording ship activity and to ensure safety. A ship's AIS transponder (the device that sends and receives signals) uses a very high frequency (VHF) RF transmitter to broadcast important information to receiver devices on other ships or land-based receivers. The information includes positional data displayed on radars along with ship details and metadata. Positional data also includes the course, rate of turn and speed of a ship. Secondary data or metadata will include information such as destination, ETA, type of vessel, ship contents and name. AIS transponders integrate a standardized VHF transponder with a positioning system, such as a long-range navigation system or global positioning system (GPS) receiver, with other electronic navigation sensors, such as a gyrocompass (Smith, O'Keeffe, Aldous, & Agnolucci, 2013). By maintaining a standard protocol for communication internationally, vessels can maintain communication across national borders.

2.2.2. Regulations and Procedures

In 2004, the International Maritime Organization (IMO) implemented laws regarding the mandatory use of AIS transponders aboard most vessels. The Safety of Life at Sea (SOLAS) Convention, Chapter V, states: "All ships of 300 gross tonnage and upwards engaged on international voyages and cargo ships of 500 gross tonnage and upwards not engaged on international voyages and passenger ships irrespective of size shall be fitted with an automatic identification system (AIS)."

AIS devices do not send constant signals regarding position; instead, signals are sent at specified intervals. By automatically sending messages, ships can ensure safe course and avoid collision without the need to see other vessels. AIS communication is based on time division multiple access (TDMA) systems. TDMA is a channel access method used in networks. With multiple devices communicating on the same channel, TDMA allows for devices to use time slots and share the same transmission medium (e.g., radio frequency channel). Self-organizing TDMA (SOTDMA) is the system responsible for maintaining order when transmitting signals between devices. With SOTDMA, devices must declare what time slots they will use when transmitting signals to avoid interference. This allows AIS devices to organize communications to optimize efficiency. Depending on where the vessel is located and the speed of travel, the SOTDMA protocol will adjust the time interval. Time slots can vary from less than 3 seconds, when vessels are traveling fast or changing course, to 3 minutes, when they are at anchor. Figure 2.1 depicts the use of time slots.

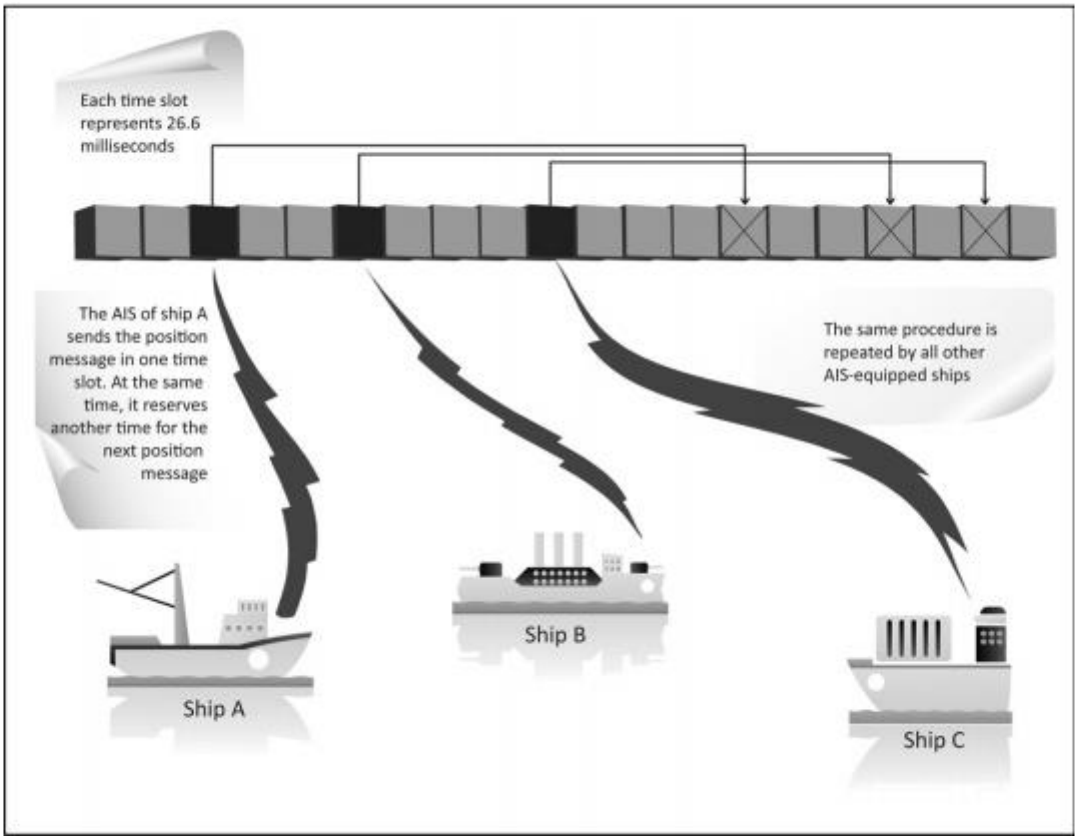


Figure 2.1: SOTDMA protocol for reporting timeslots. Variable timeslots exist depending on the ships sped and where it is going. (Ball, 2013)

Multiple classes of AIS devices exist and vary based on the vessel's requirements. For this research, all data is limited to Class A vessels. Class A vessels are 300 gross tonnage and upwards, and engaged on international voyages (Ball, 2013). Almost all cargo ships travelling across international waters fall under the IMO's regulations regarding mandatory use of Class A AIS devices.

AIS data is limited to approximately 50 nautical miles due to the curvature of the earth. Satellite-based Automatic Identification System (SAIS) solves this issue by using satellites to receive and transmit AIS signals. Moreover, it allows for the collection of data from many vessels across large distances. This is useful for big data applications as messages from thousands of vessels can be sensed by one satellite, thereby being more cost effective and greatly improving the monitoring of vessel traffic patterns and identification of potential threats (Ball, 2013).

2.2.3. Study Region

The dataset used in this research was centered on the eastern side of the north Pacific Ocean. The Port Metro Vancouver (PMV) was selected as the research area for model development and evaluation. Port Metro Vancouver is the largest Canadian port by tonnage and the fourth largest in North America. Cargo ships proved ideal for analysis as the majority of traffic is shipping-related and fall under IMO regulations (19 of SOLAS Chapter V: *mandatory AIS devices on all large shipping vessels and all commercial passenger vessels*). On January 1, 2008, the Fraser River Port Authority, North Fraser Port Authority and Vancouver Port Authority combined to become Port Metro Vancouver. Positioned on the Southwest coast of British Columbia, PMV covers more than 600 kilometres of shoreline and extends from Point Roberts at the Canada/U.S. border eastward to the Fraser Valley and includes the North and middle arms of the Fraser River. Bordering 16 municipalities, PMV works with elected officials, city staff, residents, and businesses to balance the needs of the shipping and tourism industries and local communities. The Port is committed to sustainable operations and development and mindful of economic, social, and environmental impacts (*Port Metro Reference Guide*, 2016).

2.3. ABM Software

The use of Agent-Based Modeling (ABM) as a form of research has been growing in several fields. As a result, many ABM toolkits have been developed for a variety of applications. While each toolkit has a variety of characteristics, there is a common set of criteria they all follow. Serenko and Deltor (2003) provide a summary as to why ABMs toolkits are useful:

- provide abstractions in which programmers can build from
- incorporate features of visual programming, which saves time and makes development easier, more attractive, and enjoyable
- offer run-time testing and debugging environments
- allow programmers to reuse classes (definition of objects) created by libraries or other programmers (Serenko & Detlor, 2003).

Several toolkits are popular amongst researchers. Toolkits can be categorized into two main categories, the first of which follows the “framework and library” model. Examples include AnyLogic, Ascape, MASON, Swarm and Repast. This category of software is built with a set of standard concepts for designing ABMs along with a library of simulation tools used for modelling. This differs from the Logo family of models, most notably NetLogo. This group of software aims to provide a high-level platform that allows for a wider range of applications. NetLogo was designed for simple and rapid model development, but has, over time, developed into a sophisticated modelling platform with many capabilities that the framework and library models also contain (Railsback et al., 2006).

2.3.1. NetLogo Overview

NetLogo is a free and open-source ABM programming language and integrated modeling environment. Authored by Uri Wilensky in 1999, it has been in continuous development at the Center for Connected Learning and Computer-Based Modeling at Northwestern University (Hong Zheng, 2013). It is designed for both education and research and is used across a wide range of disciplines. NetLogo runs on the Java virtual machine; thus, it works on all major platforms (Mac, Windows, and Linux) and runs as a standalone application, or from the command line. NetLogo also provides a classroom participatory-simulation tool called HubNet. Models and HubNet activities can be executed as Java applets in a Web browser. NetLogo was modelled on the Logo programming language and aimed to have a low threshold to entry requiring less programming knowledge. Although the primary purpose of NetLogo has been to provide a high-level platform allowing users to build and learn from simple agent-based models, it now contains many sophisticated capabilities.

Railsback et al. (2006) commented that NetLogo is suitable for developing models that are compatible with its paradigm of short-term, local interaction of agents and a grid environment, and not extremely complex. It is even recommended for developing prototyping models that may be implemented later by using lower-level platforms; starting to build a model in NetLogo can be a quick and thorough way to explore design decisions. Its intermediate execution speed may not be a significant limitation for many applications, especially compared with the potential reduction of programming time. On one hand, with its heritage as an educational tool, NetLogo stands out for its ease of use and excellent documentation. On the other hand, its simplified programming environment restricts experienced programmers when making a detailed or large-scale model. For instance, it requires having all code in one file and enforces less organizational discipline than is required in Java or Objective-C and thus can be cumbersome for large models.

NetLogo is built around the environment interface and the code section. The code section is one area where users can program their models in addition to a command center dialog box present on the environment interface. The environment interface includes most importantly the "world" box or a 2d space divided into a grid of patches. The environment interface is also where users can add a variety of built in "buttons" or inputs where variables can be defined, represented, or adjusted for use in the simulation. The world box is populated with different agents than can follow a variety of instructions.

In NetLogo there are four types of agents: turtles, patches, links, and the observer. In relation to CA, turtles represent mobile CA cells, patches are CA cells, links are aggregated to the CA cells (turtles or patches), and the observer acts as an all knowing “god” that can dictate behaviours and report on CA states. Each patch agent is a square cell in the world box that bears unique coordinates. The patch at coordinates (0, 0) is called the origin and the coordinates of the other patches are the horizontal and vertical distances from this one (i.e., a local planar coordinate system) with coordinates denoted pxcor and pycor.

The world of patches can be unbounded and allow turtles to move past the edge of the world and appear on the opposite edge (i.e., similar to a plane mapped onto a torus). Turtles are agents capable of moving around the grid of patches. Turtles also have coordinates which represent the patch they inhabit. Turtles have a unique identifier called "who" which allows them to identify themselves or other turtles in the simulation. Links are agents that connect two or more turtles. The observer is the final agent; however, this agent does not occupy any space on the world. Instead, the observer gives instructions to other agents or communicates directly with the command center serving as an output for various information.

When NetLogo starts up there are no turtles, the observer can create turtles or patches can "hatch" their own turtles. With the use of commands and reports the user can tell agents what to do. Commands are actions for the agents and reporters carry out some operation and report a result to a command or another reporter. Commands and reports built into NetLogo are called primitives. There are hundreds of primitives used to carry out a variety of functions. NetLogo also offers extensions that are created by the NetLogo team or members of the NetLogo community. These extensions contain their own list of primitives that can be useful for accomplishing tasks that the built-in set of primitives fail to cover. Commands and reports that are user defined are called procedures.

2.3.2. Variables

Agent variables are places to store values. Each agent type has its own variable class. With a global variable there is only one value to the variable and all agents can access this value. An example of a global variable would be something used in many parts of the program such as time. All other agentypes (turtles, patches, and links) differ as each agent has its own unique value. For example, every patch has coordinates that are different from one another. Some variables are built into NetLogo for instance all patches have a x and y coordinate, a color value, a label, and a label color. All turtles have a who value (identifier), color, heading, x and y coordinate, shape, label, and label color. There can be many different types of turtles in a given model. The user can define each type of turtle by creating a different breed. The breeds are defined using the breeds keyword, at the top of your model, before any procedures. Each breed can have "breeds-own" variables which are unique user defined variables that are only common to that specific breed. Using the "set" and "let" commands the user can define specific variables. Set is a global variable and is applied and stored not only in the procedure it belongs to but for the entirety of the model. "Let" is for local variables and used only in the context of a particular procedure or part of a procedure. If the user applies let at the top of a procedure, the variable will exist throughout the procedure. If you use it inside a set of square brackets, then it will exist only inside those brackets.

Besides breeds, an agentset is a set of agents the user can isolate to perform a specific task. Agentsets can contain any kind of agent but no more than one type (turtle or patch, not both). The user can construct agentsets that contain only some turtles or some patches. For example, all the red turtles, or only the patches with an x coordinate equal to one.

2.3.3. Ask & Context

NetLogo uses the "ask" command to give commands to turtles, patches and links. When using the ask command, all code that is asked of the specific agent must be in the correct context. Context is set in one of three ways; With a button, by choosing the agent type from the popup menu, in the command center by choosing the agent type from the menu, or by following the ask command with the name of the agent the user wishes the procedure to be applied to.

2.3.4. Tick Counter

In NetLogo models, time passes in discrete steps, called "ticks". The built-in tick counter (location) is above the world box and keeps track of how many ticks have passed. Using the tick command, the user can identify when the tick counter will increase according to a certain action in the code. For example, increase the tick by 1 every time "x" turtles move. You can also reset the ticks when necessary, using the reset-ticks command.

2.3.5. Lists

In the simplest models, each variable holds only one piece of information, usually a number or a string. Lists let the user store multiple pieces of information in a single value by collecting that information in a list. Each value in the list can be any type of value: a number, or a string, an agent or agentset, or even another list. Lists allow for the convenient packaging of information in NetLogo. If agents carry out a repetitive calculation on multiple variables, it might be easier to have a list variable, instead of multiple number variables. Several primitives simplify the process of performing the same computation on each value in a list.

2.4. Model Description

2.4.1. Setup

Loading extensions and setting global- and turtle-specific variables are the first steps in creating a model in NetLogo. Extensions allow NetLogo to load user-created commands that can be written in Java or other languages. Users can create their own extensions or find extensions created by NetLogo or members of the NetLogo community. These extensions provide a variety of new functions not included in the default set of NetLogo primitives. Many users import extension libraries for abilities that do not exist in NetLogo or to provide better performance when accomplishing specific tasks. The MSRM requires the use of the GIS extension for loading raster imagery, applying elevation values and exporting appropriate files.

The next step is "breeding" the different kinds of agents. Each breed has its own set of variables that behave in different ways.

2.4.1.1. Borders

Borders represent any patch in the model with elevation less than 0 and a neighboring patch with elevation greater than 0. This breed is used to populate areas of the map where water borders land and is useful when creating a threshold for ships when trying to avoid land during route selection. With the ability to locate border patches, the cost of patches within a certain distance of the "border" can be increased efficiently and can ultimately alter the path selection to avoid unrealistic trajectories.

2.4.1.2. Waypoints

Waypoints are the agents created to represent the starting point for all ships in the simulation. They will "hatch" vessels in the model. Hatch is a primitive in NetLogo used to instruct a specific patch or agent to create a turtle (agent). Waypoints also contain information that will be given to any ships they hatch. This includes the speed of the ship and the fuel variables. Waypoints are created in two ways: 1) through the ship-source input on the environment interface (Figure 2.2 -Netlogo interface (see red dot 1)), which is executed by the import-elements function in the draw-map procedures, or 2) by using the place-item button on the environment interface (Figure 2.2 -Netlogo Interface (see red dot 2)).

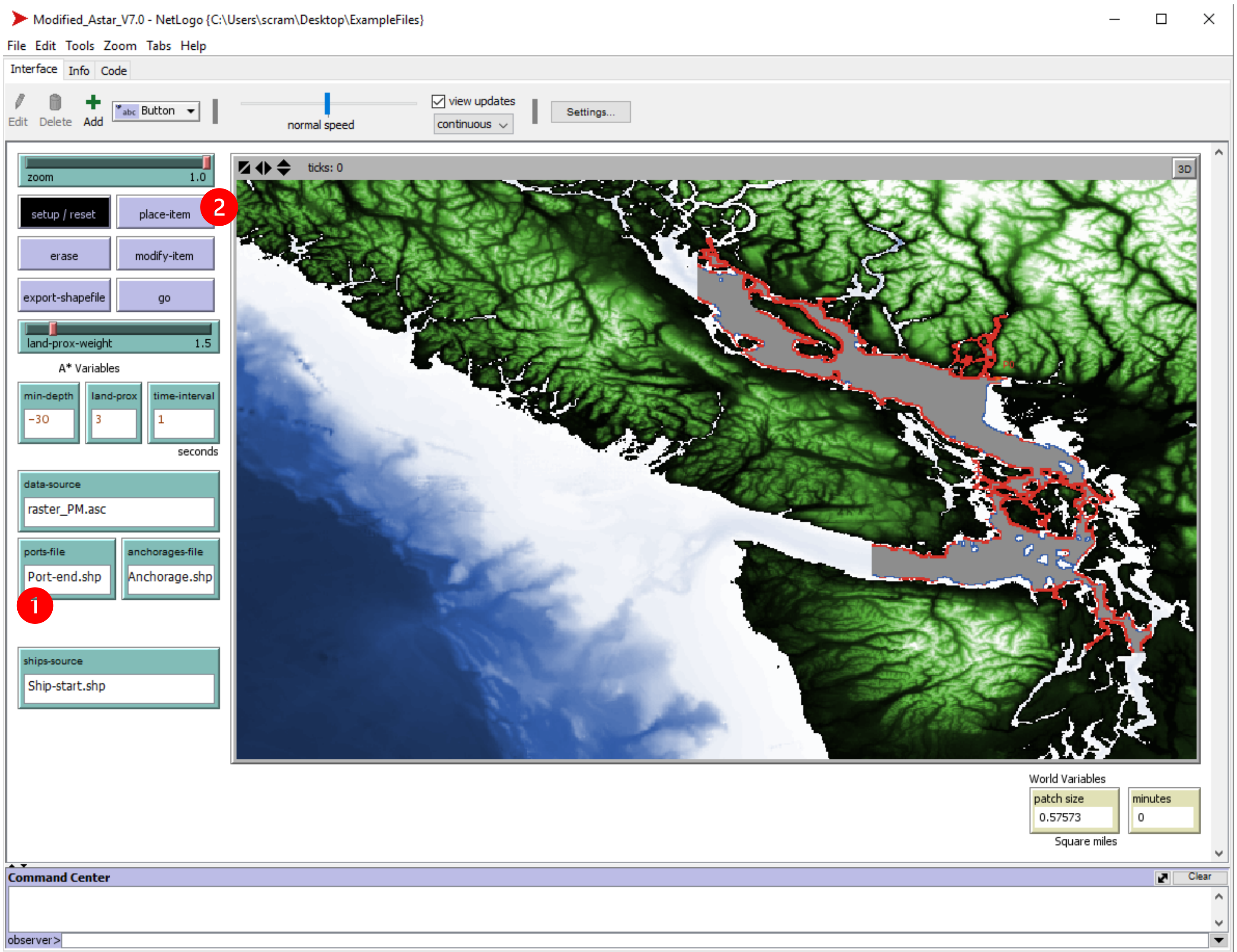


Figure 2.2: Netlogo interface with icons indicating functions of interest.

2.4.1.3. Ships

Ships is the agent responsible for traveling the list of patches from source to destinations and creating footprints along the way. Ships contain all the information used in path finding and some initial variables set by their waypoint or starting location. Some of the path finding variables include the current patch, the previously visited patch, the time interval, and the total time. Some of these variables are Boolean and only used to identify states of the ship (reached destination etc.), while some are variables responsible for correct movement. For example, with multiple anchorages and ports, each patch will obtain a separate value for its cost associated to each destination. The ship will have information on its destination (or current layer) and will choose the correct path for each destination. Since the ship can have multiple destinations, once it has arrived at its first destination, it will select a new destination along with the layer or cost for all the patches associated with the new destination. Ships are also responsible for creating footprints every time it reaches a new patch. This is necessary for exporting the data and to have a permanent history of the ship's movement and variables.

2.4.1.4. Anchorages

Anchorages are the destinations in the model. There can be many anchorages on the path of a given ship. The final anchorage of the model is known as the port. Once the ship reaches the port, the simulation is over. There can be many ports and anchorages, each created in the same fashion as waypoints – that is, using either the input (ports-file, anchorages-file) or place-item button on the environment interface. Anchorages include information on the wait time and wait list. As the ships arrive to anchorage, they will occupy a spot on the list (i.e., the first ship occupies spot one and so on) and depending on the wait time dictated by the port, it will hold the ship until the wait time has expired. This is to ensure an orderly progression when entering port and is crucial for experimenting with anchorage wait times.

2.4.1.5. Banners

Banners are the function required to place labels on the waypoints while being able to control the banners location, size, and style. The banner acts as an agent that is linked to a chosen anchorage. This agent is invisible but contains a label allowing it to be at any angle or distance relative to the agent. This allows for more flexibility when creating labels based on the scale of the map and the size of the waypoint. You can create a separation from the destination and place the label at any angle, which allows the user to view the model without obstruction.

2.4.1.6. Footprints

Footprints represent the path a ship has taken to reach the destination. They are created once a ship has reached a new patch. Footprints store all the relevant ship data including ID, time, heading, speed, and fuel consumption. The footprints are useful for multiple purposes - they act as the dataset that is exported as a point shapefile and can be analyzed elsewhere and they provide a history of all the locations the ship has previously traveled. This is crucial when telling ships not to return to their previous location and plays a crucial role in the move function.

2.4.2. Draw Map

After initializing the variables in the model, the patches are given elevation values and the map is rendered. 'Elev' is the name given to the elevation value of each patch in the model. This value is drawn from the UTM raster file that is loaded as a data source with the GIS extension. A variable called elevation is given to the data series in the file and the world envelope is set to "elevation". This copies values from the given raster dataset to the given patch variable, resampling the raster as necessary so that its cell boundaries match with NetLogo patch boundaries. Applying color to the map greatly increases the visual appeal and makes the simulations easier to interpret. By setting a scaled color for blue below sea level and green above sea level, the model begins to resemble a map. This section includes the code for importing the model elements (import-elements). This is where data from the input buttons on the environment interface are loaded into the model.

2.4.3. Create

The create section is for placing all the agents on the map and initializing the variables. It includes the code for placing and modifying the ship and anchorages.

2.4.3.1. Place-Item Button

Once selected, the user can place a point on the map by clicking on any location in the "world" with the cursor. The place-item button is set as a forever function, which allows the command to run its code repeatedly. This is necessary as a "once button" only applies the code once and stops. This does not allow the user to select a point on the map because once the button is selected, the command is applied, making it impossible to simultaneously select a point on the map. With a forever function, it repeats the code until it is told to stop, giving the user as much time as needed to select a location. One downfall, however, is that if the mouse button is held down, many waypoints will be created on any patch the cursor is on. To prevent the creation of multiple unwanted waypoints, the place-item function stops after it has created one item, allowing the user to select the button again and place another item. Due to the order of operations in the model, the waypoints can be created after calculating the cost for all patches. Furthermore, since the cost for each patch is already stored, an infinite amount of waypoints can be created without much of an increase in computational time. The options for placing an item include an anchorage, port, ship and an obstacle. Obstacles are unique as they must be added before other objects as they change the BFS values during the path finding section of the model.

Obstacles act as an “unreachable” node and are crucial to ensure the algorithm can navigate waterways with a variety of obstacle shapes, sizes, and positioning.

2.4.3.2. Modify-Item Button

The modify button is used to change certain variables of the ship or port without having to reset the simulation. This is useful for adjusting the speed and fuel parameters of the ship, allowing the user to output many model iterations with changes in speed and fuel consumption but with the same path finding variables. This saves time and allows the user to compare changes in speed and wait times.

2.4.4. Path Finding

The path finding section uses a modified BFS path finding algorithm to calculate the shortest path with obstacles. Figure 2.3 provides a diagram describing the broad functionality of the MSRM.

Maritime Ship Routing Model Concept Map

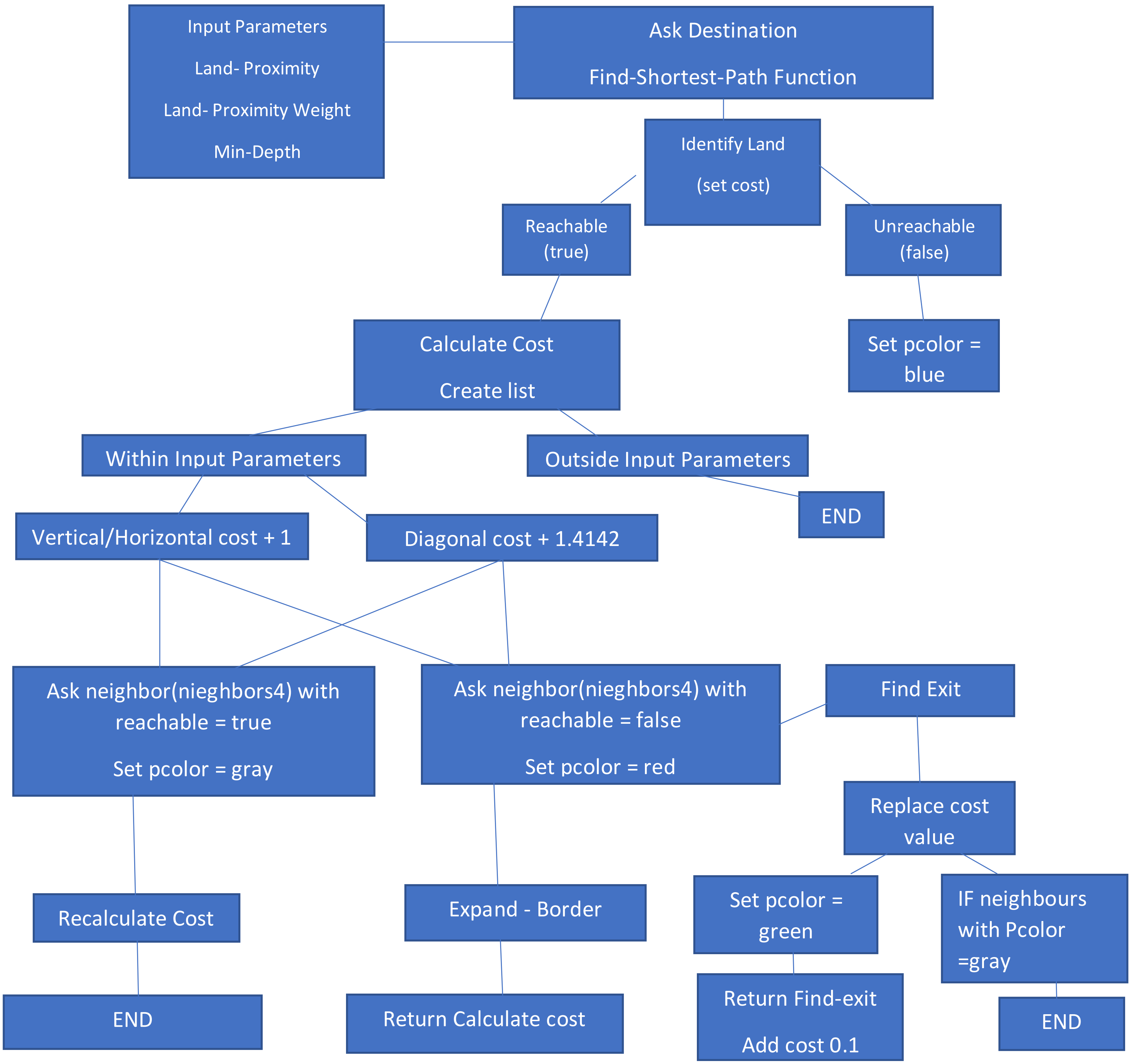


Figure 2.3: MSRM concept map. Including all input parameters and various outcomes.

The BFS algorithm provides a basis for determining the shortest path in the MSRM; however, in the presence of obstacles, modification is necessary. When ships are travelling at sea, many obstacles such as islands or ground must be avoided for safe travel. With the BF algorithm, obstacles are recognized but the path selected will always choose nodes with the smallest cost, and these nodes will be adjacent to the obstacle. This is visible in Figure 1.3 graphical representation of BFS and A* search algorithm. For maritime traffic simulation, this representation is not accurate as ships will avoid obstacles with caution and adjust their trajectories accordingly to maximize efficiency while safely avoiding any obstacle. By identifying the obstacle boundaries using the border agent initialized in the setup section, modification to the algorithm can be applied to replicate maritime traffic more realistically. This modification will essentially change the cost of nodes surrounding obstacles using several of the inputs on the interface tab. These include Min-depth, the minimum elevation required for a given node to be considered an obstacle, Land-Prox (LP), the distance in nodes from the obstacle or border that will be affected by the cost modification, and Land-Prox-Weight (LPW), the value used to multiply the cost of any nodes that fall under the LP distance.

The BFS path finding algorithm typically begins calculating values from the start of the path. In the MSRM simulation, the algorithm calculates values from the destination. Each anchorage in the model can be used as a destination and calculates its own shortest path values. By using the anchorages as the starting points in the path finding algorithm, all nodes in the map will acquire values corresponding to each destination. These values will act as a list when a ship is selecting its path to a given anchorage. By using the destination as the starting point for the simulation, it allows for ships to be added at any location on the map without requiring a new path finding algorithm to be executed as each cell in the waterway contains BFS values pertaining to all other anchors or ports. This allows for many ships to be added to the simulation without additional processing. Calculating path finding values is the most time consuming and computationally demanding process of the model. By starting from the destination, it avoids having to calculate path values when a new ship is added to the simulation, allowing the user to test many simulation scenarios without increasing the execution time required to determine shortest path.

The path finding section begins by asking each of the destination agents (anchorage or port) to execute the 'find-shortest-path-to-ships' function. This is initialized by identifying the patch it inhabits and determining if it is on land (true or false) and whether it is "reachable". Reachable is a variable used for patches that may not be on land but whose elevation is greater than the min-depth value. The next operation is the 'calculate-costs' command that starts by creating a list for the nodes in the map. Each list entry is increased by adding the value of either 1 or 1.4142 each time the operation is executed. If the node occupies a space that is vertical or horizontal from the last node, it will add a value of 1. If the node is diagonal, it will add a value of 1.4142 because the length of the diagonal distance of a square is $\sqrt{2}$ or approximately 1.4142. Since a ship can travel diagonally, the values must accurately represent the difference is distance. These nodes are colored grey for the duration of the path finding algorithm to help visualize the procedure.

Recalculate cost is the function that modifies the A* algorithm by recalculating the cost of any node that is within the predetermined distance to land (LP) and increases the value. This function asks any node that is not land and is adjacent to a border patch to be selected if they are within the distance determined in the LP input on the interface. These nodes are colored red - not only to help visualize the pathfinding portion of the simulation, but also as a way of identifying the difference between patches that do not fall within the LP boundary (grey nodes). The red nodes have their cost altered according to the following formula

$$Recalculate\ Cost = Cost + (LPW * (LP - distance\_to\_land)\ /LP)) \tag{3}$$

Distance-to-land is the minimum distance to the closest border patch (in cells). By subtracting the distance-to-land from the LP value and dividing the sum by the same LP value, cells further from the border will have their cost adjusted less than cells closer to border agents when multiplied by the LPW. At distances beyond the desired LP the maximum value is subtracted from the shortest path scores. Without this formula, all patches within the LP will have the same cost increase. This is crucial for BFS modification to produce as realistic shipping routes as possible. Furthermore, when the LP is large enough to affect many or all cells in a given waterway, this modification will ensure the path created will still prioritize avoiding obstacles. These operations are executed for each new patch on the map using the expand border function to continue selecting new nodes.

The final operation used for path finding is the ‘find-exit’ function. The ‘find-exit’ function is used to further recalculate values for any node that has been colored red. This procedure is very important for proper functionality of the simulation. After the recalculate cost function, path finding values for the red nodes are much larger than the nodes that are not recalculated. This can create a scenario where a ship can get “stuck” and fail to reach the destination. When a ship is selecting the next node in a path, it chooses a neighbouring node that is reachable with the smallest cost. Since the recalculate cost function increases the values for certain nodes, it will change the original path finding values and many nodes closer to the destination may no longer possess the smallest cost of all adjacent neighbours. A neighbor that is farther away from the destination may have a lower cost because it is not increased by the recalculate cost function as much or at all. In this scenario, the ship will fail to reach the destination and repeatedly visit the nodes with smaller costs around the destination and the simulation will continue to run until the user is forced to stop the program. The 'find-exit' function starts from the destination and will only change the cost of nodes that are red, stopping once it has reached a node that is grey (outside the land boundary). It will change the color of the node to green to make sure it does not select any of the previously used nodes and so that it finds a correct path to open water (grey nodes). Since it uses the color of the node to verify if a node can be used as the next selection from the list, changing the color of the node in the 'find-exit' function is an easy way to identify whether nodes have been used in the past, preventing the ship from failing to find an exit.

2.4.5. Movement

All potential destinations have populated lists for path finding values and the ship agents are ready to use the data for movement. This part of the model includes all the functions required for the ships to move from any node on the map to the destination. This part of the model is recursive, similar to the path finding algorithm functions. However, the tick counter is now utilized to determine how many iterations of code have been processed. This is the first line of code and will count every time the ship moves. The ticks are used as a time and total time function in the model as one tick is set to the time-interval (in seconds) input on the interface. This means that for each second of “real time”, the ship will execute one iteration of the move function. This is done by using the scale of the map and changing the distance the ship travels in each tick or iteration of the code. By determining the speed of the ship in knots (selected with the input dialogue box when creating a ship), the distance the ship travels will change based on the time interval in order to create real time in the model. The move section outputs to the command center with the speed in knots and the distance the ship will travel in the time interval in ticks. This is useful for troubleshooting and ensuring the ship distance is correctly calculated for all ships in the simulation. This is also verified outside of NetLogo to ensure accuracy. The move function includes the creation of footprints. Footprints are agents that are created along the path the ship takes. Due to the modularity of the time interval and the variable speed (velocity), the ships can move a small fraction of a patch during one tick of the simulation. To avoid the ship from creating a footprint every time it moves such a small distance, the function only calls for a footprint to be created once the ship has landed on a unique patch. By creating footprints, the data can be collected in an easy way and exported for further analysis. This is shown in Figure 2.4 Model Movement.

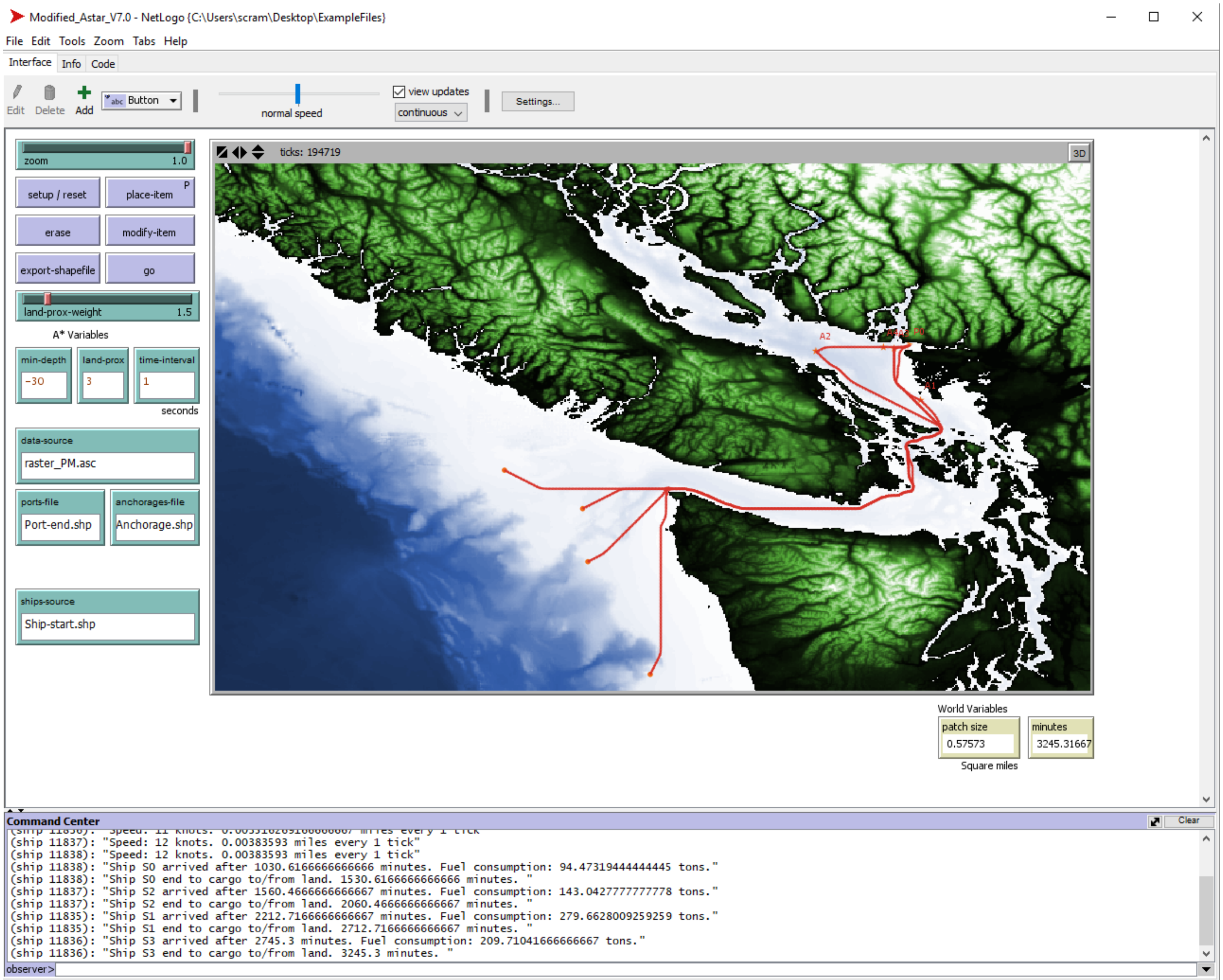


Figure 2.4: Footprints and command centre. Example of footprints created for several ship pathways and multiple anchors. Highlights information gathered in the command centre.

2.4.5.1. Fuel Consumption

Determining the amount of fuel a ship will use on a given journey at sea can be complicated. Many factors such as the ships draft, displacement, weather force and direction, hull and propellor roughness are all factors that need to be considered (Bialystocki & Konovessis, 2016). Nicolas BIalystocki et al. 2016, have proposed a prediction algorithm using statistical analysis of 418 Pure Car and Truck Carrier (PCTC) transport ships. In their analysis they were able to determine the average values necessary to provide an equation for calculating rough fuel consumption of PCTC vessels. The equation proposed is as follows:

$Fuel\ Cons\ =\ 0.1727\ \times Speed\text{^}2 - 0.217\ \times Speed$ (Bialystocki & Konovessis, 2016) (4)

The AIS data utilized for MSRM comparison consists of bulk shipping vessels including pure car and track carriers. Although the data source used has a variety of bulk cargo vessels, the same assessment can be used as a rough estimation on the amount of fuel consumed by each ship in the MSRM. Without any parameters in the MSRM regarding weather parameters, ship displacement due to currents and changes in speed or rutter positioning determining accurate fuel usage is not possible. By using and average calculation for all simulations, a comparison amongst ships in the model can be assessed. This may provide useful for anchorage scheduling analysis and further implementation of the MSRM. The code for implanting the calculation is as follows:

$$set\ fuel\_cons\ (((A\_fuel\_cons\ *\ speed\ *\ speed) - (B\_fuel\_cons\ *\ speed))\ /\ (24\ *\ 60\ *\ 60))\ *\ totaltime \quad (5)$$

Where fuel-cons is the final estimation on fuel used in tonnes per segment. A-fuel cons and B-fuel cons are input during ship creation and the same value is used for all model iterations in this study. The total time using the standard 24-hour clock is calculated to present the fuel consumption value as a function of time as opposed to number of segments, or ticks in Netlogo. The ability to change the A and B fuel cons parameters can be performed to compensate for experimentation with different ship types, changes in weather etc. This may provide useful for further implementation of the MSRM as different A and B fuel cons parameters can be used to examine changes in various weather/current patterns or ship parameters.

2.4.6. Other

The final section includes extra functions that are used at the end of the model. The first is determining the miles per patch of the map. Miles are used instead of kilometers due to the formula used to determine the fuel consumption of the ships. This section also includes applying the haversine formula to account for the curvature of the earth. Without this calculation distances, velocity or ships, time and as a result fuel-consumption calculations would not be accurate. The GIS plugin in Netlogo ensures correct projection and distance measurements when exporting shapefiles to account for the curvature of the earth. However, the metadata in the footprints created by the ships would not calculate correct fuel or time parameters without implementing the haversine formula into the miles per patch calculations. The calculation is as follows.

$$Miles\_per\_patch = (item\ 1\ world - item\ 0\ world) / (max\_pxcor - min\_pxcor) * pi / 180 * earth\_radius \quad (6)$$

The world items are used to determine the scale of the map which in turn changes the size of the nodes. This is divided by the max extent of the world to determine the exact distance per node regardless of the scale of the map. The result is multiplied using the haversine formula to calculate the distance between two coordinates on the earths plane (pi/180 * earth radius which is 3959 miles). The unit of measurement is changed back to kilometers in QGIS during the analysis.

The next function is the ‘export-shp’ command. This is used to name the files as they are exported. The files are named after the LP and LPW for each model iteration. These names were selected for easy comparison when analyzing the results. By labeling the shapefile before importing to other software, it reduces mistakes when identifying each model iteration.

2.5. Data collection

2.5.1. Model iterations

To verify the accuracy of the MSRM, results were collected and compared to a large AIS dataset that included data from 2012 of ships travelling across the Pacific. Several iterations of the model were selected for comparison. The values for land-proximity (LP) and land-proximity-weight (LPW) were used to distinguish and name each iteration of the model. The changes in these values alter the calculations of the algorithm, generating different paths. The amount of change is directly related to the increase in LP and LPW values. By using PMV as a case study, AIS data was used to compare different model iterations for path likeness.

Careful selection of model iterations was performed to help illustrate the changes in the path finding algorithm and how they affected the model results. The iterations included LP values of 5, 3 and 2. Other LP values were tested in model development however, the upper and lower range of values (5, 2) provide the extent to which the modified BFS algorithm can correctly perform. Applying LP values larger than 5 does not function well with such a congested waterway, as some channels in the Port Metro region are narrower than 10 patches. As a result, all the cells will have BFS costs modified, traversing farther from islands, preventing the model from selecting the desired path. LP Values below 2 do not provide enough modification of patches, creating paths like an unmodified BFS algorithm. An example of this is analyzed using model iteration LP2 LPW2, where the low weight combined with smaller LP creates incorrect path replication. This is covered in more detail in the results section. The functionality of the MSRM does not allow for much variation in LP due to waterway restrictions and the significant influence larger values have on simulation functionality. Smaller deviations in LP weight (0.1-0.5) were tested during development and although they provide interesting results, narrowing down selection was necessary for assessment.

The LPW values selected for analysis have a greater range, with a maximum of 5 and a minimum of 1.5. These differ more than LP values as they do not drastically change route selection and slight changes in LPW provide more detailed pathfinding, useful for analysis. As with LP, upper and lower boundaries limit the range to which LPW values can be selected. The lower range (2) provides minimum change to patches in the LP distance, and LPW values smaller than 2 do not change route selection sufficiently in the Port Metro scenario. The upper range of LPW values with low LP values will reach a limit and increasing LPW beyond a certain point will not change the path. Larger LP values in a different waterway scenario would allow for higher LPW values, but still a limit exists where route selection will not change as the shortest path may not be within the LP range and such large changes in LPW values have no effect. In most port situations the LPW limit will be rather low (<10), depending on scale, as waterways are usually narrow, and navigation is more difficult.

The choice of model iterations used for analysis depend on the scale and shape of the given waterway. Different ports or maps of vastly different scales require changes to the LP and

LPW values to most accurately replicate ship movement. By exporting and comparing the results to AIS data, the pathfinding values can be optimized to replicate shipping routes of any waterway. This is exemplified with the Port Metro case study and discussed in more detail in the results section.

2.5.2. AIS data

The original AIS dataset contained a variety of vessel types and messages. To effectively analyze traffic patterns, the data was separated into categories based on vessel type, date and region. The vessels' navigation status included dynamic and static messages, but for the purpose of traffic analysis, only dynamic messages were necessary. When analyzing maritime traffic patterns, it is important to isolate regions or ports as they remain unique due to logistical and spatial characteristics (Martineau & Roy, 2011). The data was accessed through a SQL server where queries were applied to extract the necessary values. This included setting a bounding box around the region of Port Metro, pulling only cargo vessel results, analyzing results in month-long portions, and discarding any static messages. After the results were filtered, several comma separated value (one per month) files were created that included the identified criteria. The results were projected in QGIS to ensure the correct region was included and the attributes were verified.

2.6. GIS Analysis

2.6.1. Visualize & Organize

QGIS was used to visualize, organize, edit and analyze the data. QGIS is an open-source geographic information system (GIS) software where users can analyze and edit spatial information. It may lack some functionality when compared to other GIS platforms; however, like NetLogo, its accessibility and being open source provided its own benefits. Such benefits include the fact that it is freely available, it can be installed on different operating systems, it does not require expensive hardware, it is developed by different users worldwide, users have access to the source code and there is less processing time and better rendering capabilities (Tisue, 2004). For the purpose of this analysis, it provided all the required tools.

Before the presentation of data, basemaps are needed to visualize the study area. This included a raster image containing the bathymetry data for the Port Metro area (the same data used for the NetLogo raster) and a vector shapefile of North America. The data was first imported to ensure accuracy in scale and location. This was done by illustrating all the model data on the basemap and visually identifying alignment with the same coordinate system (Figure 2.5 Raw Model Data).

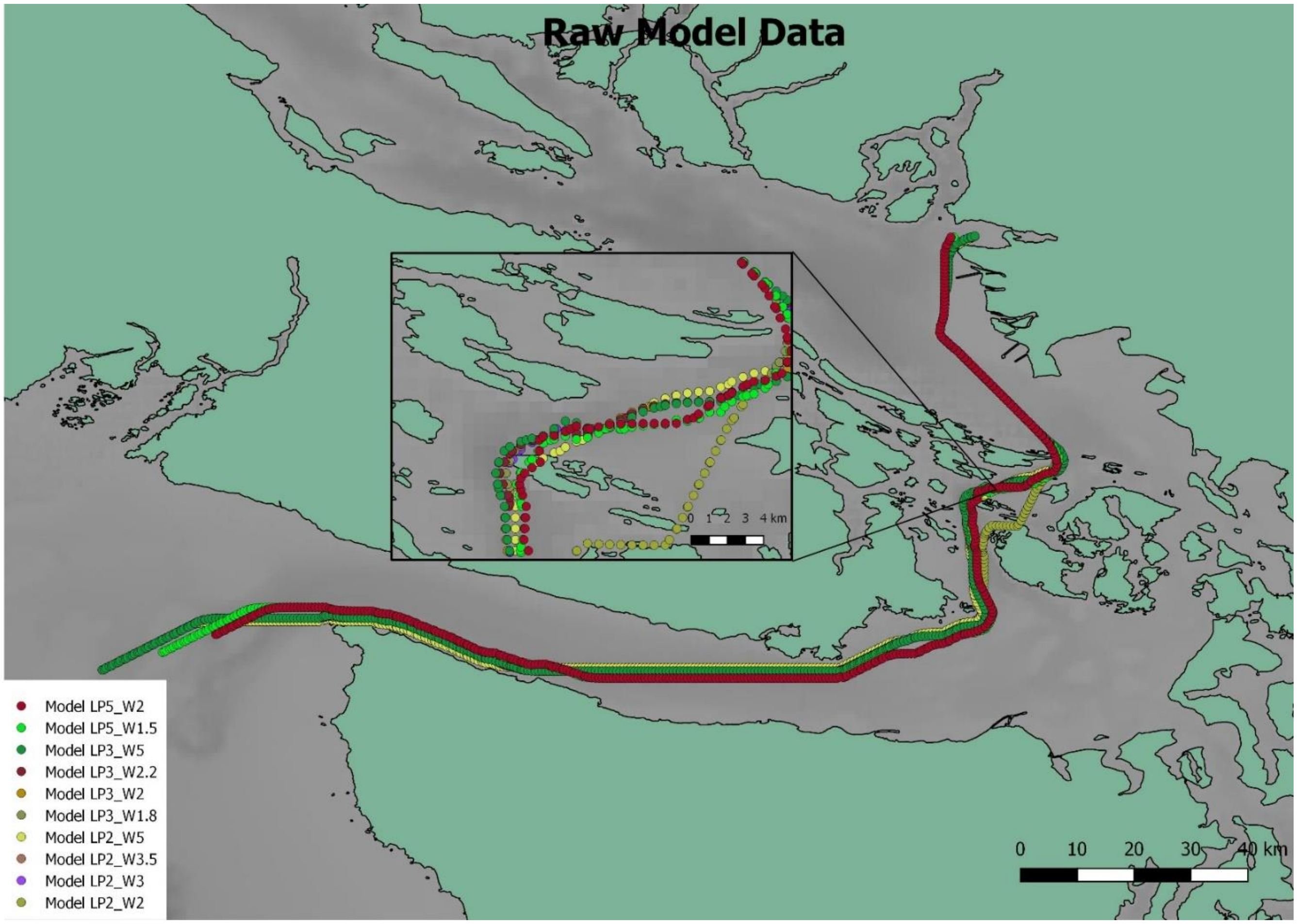


Figure 2.5: Raw model data. This is the original output from Netlogo in QGIS before any adjustments are made.

The bathymetry data consists of an average value of elevation across an entire pixel. The North America vector shapefile is not necessary for analysis but allows for better visual identification of land. However, the North America shapefile is overlapping pixels where the average elevation is below sea level. This makes some of the ship nodes appear to cover land. Having a raster image with a smaller scale improves the results and imagery; however, the computational time and power required increases significantly (Figure 2.6 Vector Overlap).

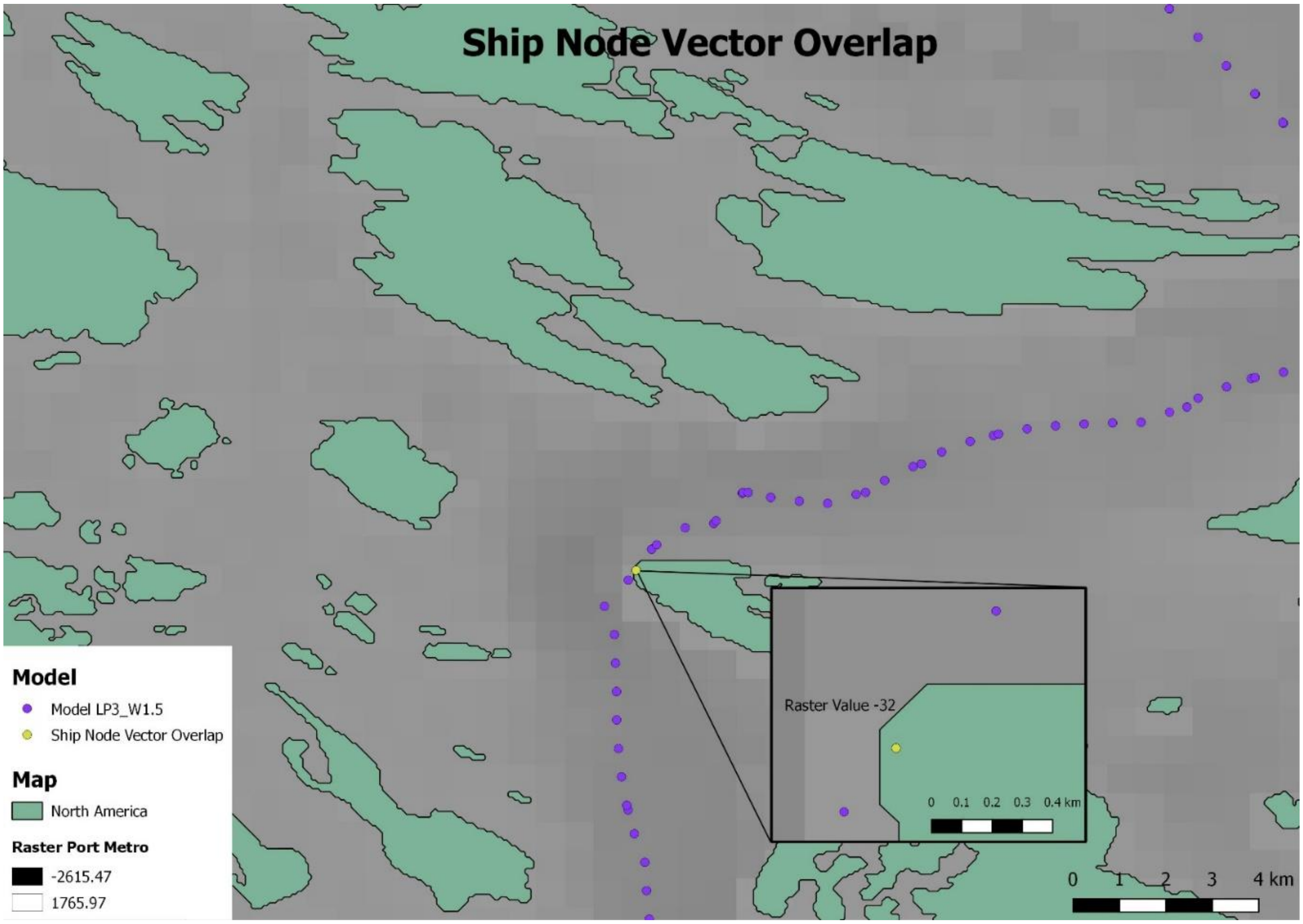


Figure 2.6: Ship node vector overlap. Due to the scale of various maps, nodes can appear on land.

The metadata was analyzed to ensure the model had exported the correct information. By opening the attribute table, all the metadata from the model can be verified (Figure 2.7 Meta Data).

	WHO	COLOR	HEADING	XCOR	YCOR	SHAPE	LABEL	LABELCOLOR	BREED	HIDDEN	SIZE	PENSIZE	PENMODE	SHIPID	TIME	SHIPSPEED	FUELCONS
1	11825	15	42.46309783912...	277.5002856184...	137.3612311876...	default		9.9	footprints	false	3	1	up	11824	112	10	0.019574074074...
2	11826	15	53.04797860857...	278.0076989570...	137.5013074999...	default		9.9	footprints	false	3	1	up	11824	209	10	0.036526620370...
3	11827	15	53.04797860860...	278.5002101479...	137.8717950980...	default		9.9	footprints	false	3	1	up	11824	320	10	0.055925925925...
4	11828	15	42.53498938440...	279.5035214193...	138.3688811162...	default		9.9	footprints	false	3	1	up	11824	529	10	0.092452546296...
5	11829	15	75.96653994912...	280.0029289261...	138.5008353111...	default		9.9	footprints	false	3	1	up	11824	624	10	0.109055555555...
6	11830	15	53.10511343662...	280.5002496828...	138.8741648973...	default		9.9	footprints	false	3	1	up	11824	736	10	0.128629629629...
7	11831	15	42.55712789802...	281.5001060000...	139.3664259370...	default		9.9	footprints	false	3	1	up	11824	944	10	0.164981481481...
8	11832	15	76.00009636503...	281.9979514162...	139.5008368015...	default		9.9	footprints	false	3	1	up	11824	1039	10	0.181584490740...
9	11833	15	53.12415127629...	282.5007835404...	139.8753443278...	default		9.9	footprints	false	3	1	up	11824	1152	10	0.201333333333...
10	11834	15	42.63856530226...	283.5032317824...	140.3744741088...	default		9.9	footprints	false	3	1	up	11824	1361	10	0.237859953703...
11	11835	15	76.20821645046...	283.9636532864...	140.5001349014...	default		9.9	footprints	false	3	1	up	11824	1449	10	0.253239583333...
12	11836	15	53.33323419317...	284.5037168405...	140.8860544364...	default		9.9	footprints	false	3	1	up	11824	1569	10	0.274211805555...
13	11837	15	42.73188796961...	285.5028176688...	141.3793297759...	default		9.9	footprints	false	3	1	up	11824	1777	10	0.310563657407...
14	11838	15	76.31564860458...	285.9472809742...	141.5001949554...	default		9.9	footprints	false	3	1	up	11824	1862	10	0.325418981481...

Figure 2.7: Meta Data. Attributes table of simulation results in QGIS.

A subsample of the data where the ship travelled in a straight line was clipped and analyzed to ensure accurate velocity of the model. The ship speed in knots was converted to kilometres and the distance of the line (36.33 km) was determined with the measurement tool in QGIS. The time variable in the metadata was subtracted to indicate the total time from the beginning of the line until the end (11686 seconds or 3.24 hours). The total time was multiplied by the kph to determine the expected distance (37.3556 km). Although the distances are not exact, the difference of 2.7% is admissible and most likely due to inaccuracies in line measurement.

A manual selection of several ship point data from an individual trip is used for ship-wise error calculations (covered in section 3.2.6 Ship Wise Errors) and exhibiting how analysis with several ship point data compares to using large AIS data sets. The data was added to the map and given the correct coordinate system. By isolating individual ships, it became clear that in much of the single ship data location points are separated by large distances. A selection of several ships was chosen to help illustrate the gaps in the data (Figure 2.8 Selected AIS Data). Due to the sporadic nature of the AIS location data and the limited metadata, analysis would prove most useful utilizing as much of the subsample as possible.

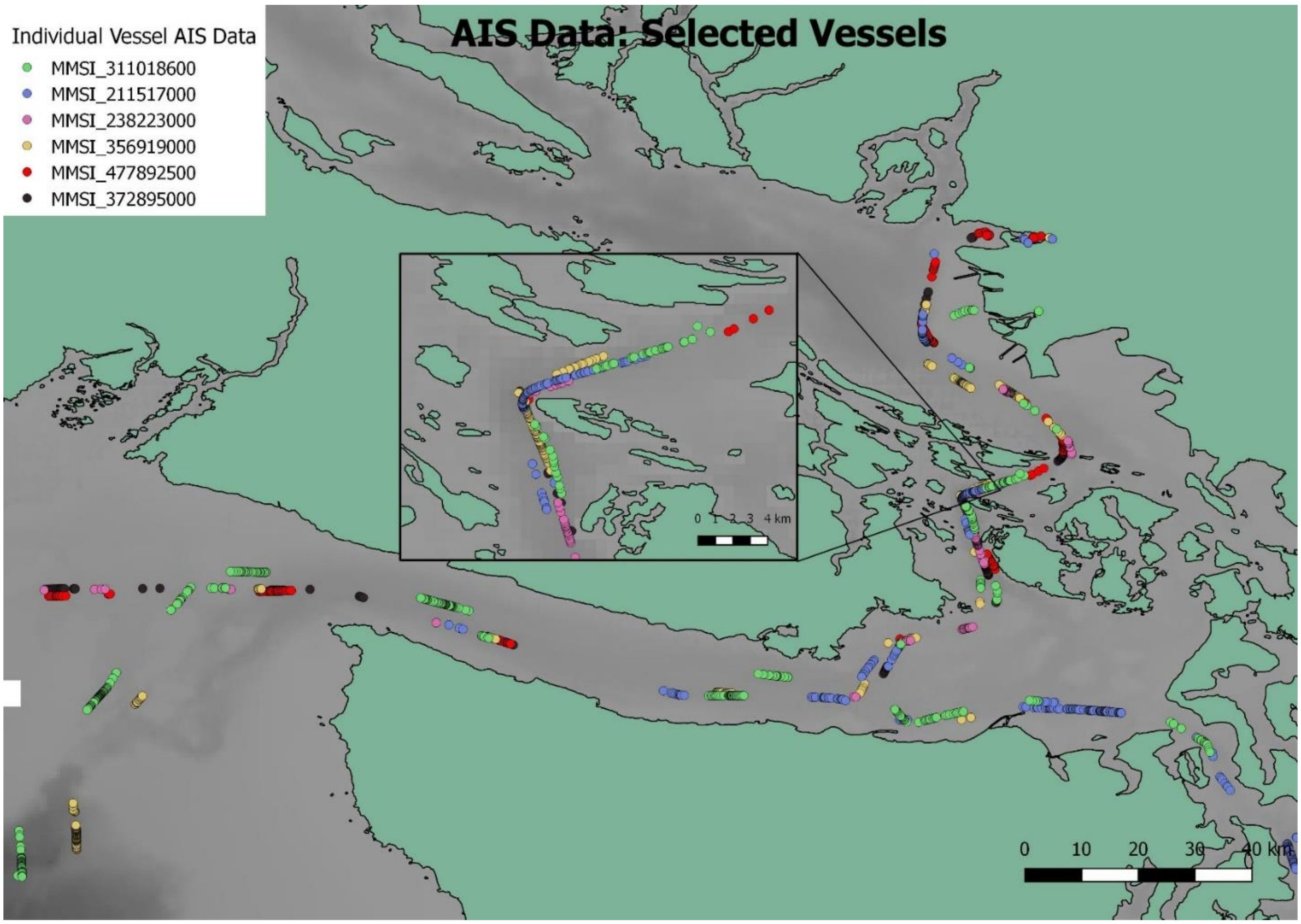


Figure 2.8: AIS Data: Selected Vessels. Manual Selection of several ships from the AIS data. Highlights the sparse nature of AIS data recordings.

Both the model and AIS data include the entire port region extending past the Strait of Juan de Fuca and into the North Pacific. However, for the purpose of comparing path likeness, a subsection of the region was chosen. The selected path, starting at the Haro Strait and continuing to Port Metro just past Saturna Island, was chosen due to the narrow channel and abundance of obstacles. Waterways with more obstacles offer the model's algorithm more choices and proves to be the most challenging to simulate. Open waters are not as strictly navigated and ships tend to navigate where weather is advantageous, or simply along predefined paths (Tam, Bucknall, & Greig, 2009). This can be observed when illustrating all the AIS data. Data existing beyond the Salish Sea and into the Pacific does not appear to follow such a defined route. Once the vessels enter the Salish Sea and the Strait of Juan de Fuca, they very clearly follow a given pathway (Figure 2.9 All AIS Data).

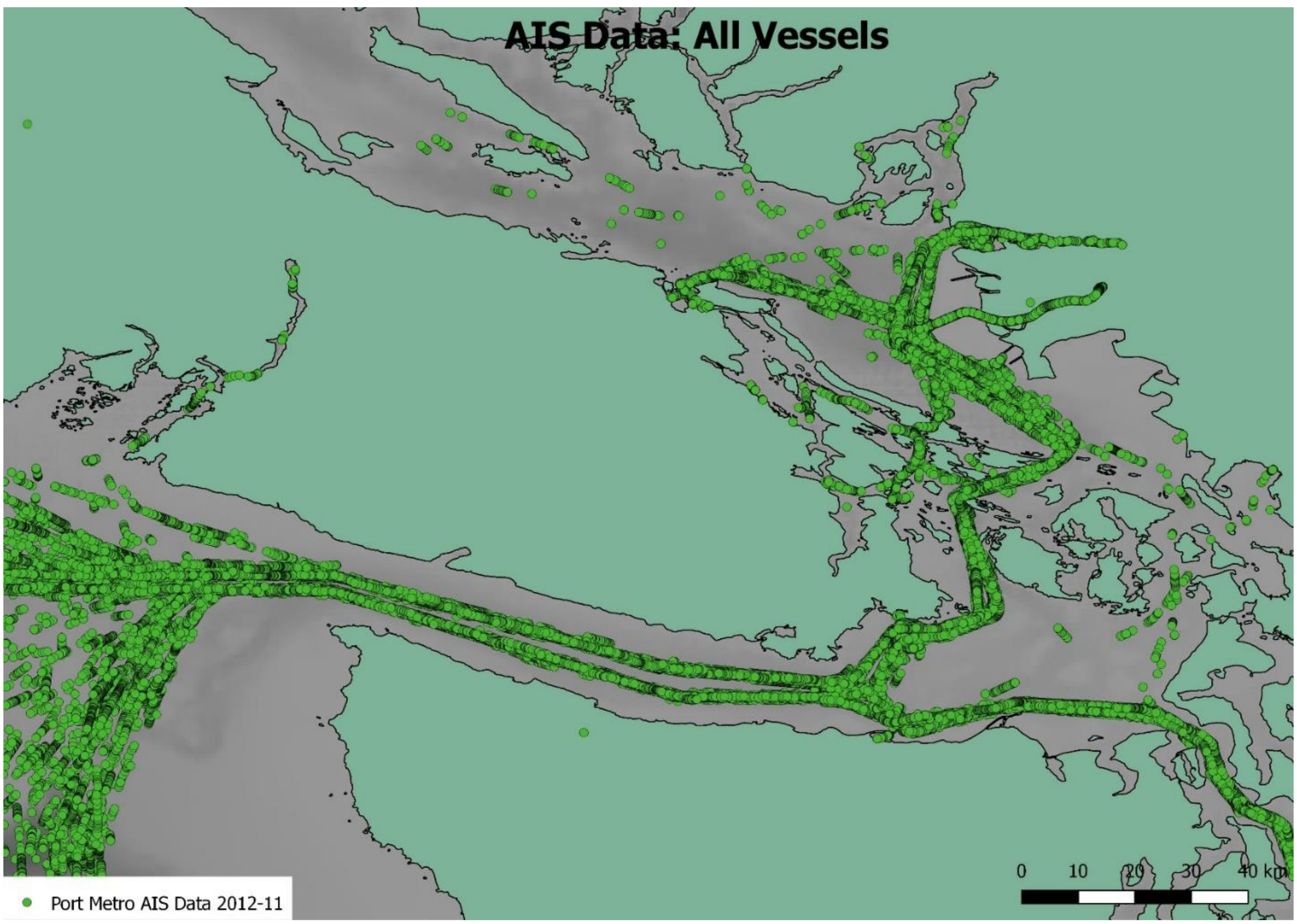


Figure 2.9: AIS data: all vessels. This shows how much data was included in the original selection of AIS messages. Clean up was necessary to make sense of the overabundance of information.

It is important to note that the nature of the model will not provide the most accurate path likeness in waterway scenarios with a large variety in size and depth. The algorithm must be modified by adjusting shore distance and weight parameters to simulate open waters or narrow channels more accurately. This weakness is described further in the Results section. By clipping the AIS data to only represent the selected region, we can more accurately analyze the relationship. Furthermore, due to the number of points in the dataset, any reduction significantly improves rendering time and prevents errors in execution. The desired location was digitized as a polygon to act as boundaries for the clipping process. The clipping tool was used to create a subset of the AIS data that only included the desired region. The region selected includes the entirety of PMV and a small amount of the Pacific Ocean where it meets the Salish Sea. It is the same location and size as the region of extracted AIS data. This process is straightforward and clips a vector layer (AIS data) using the polygons of an additional layer (digitized polygon). Only the parts of AIS data that fall within the polygons of the clipping layer are added to the output (Figure 2.10 AIS Data Clipping).

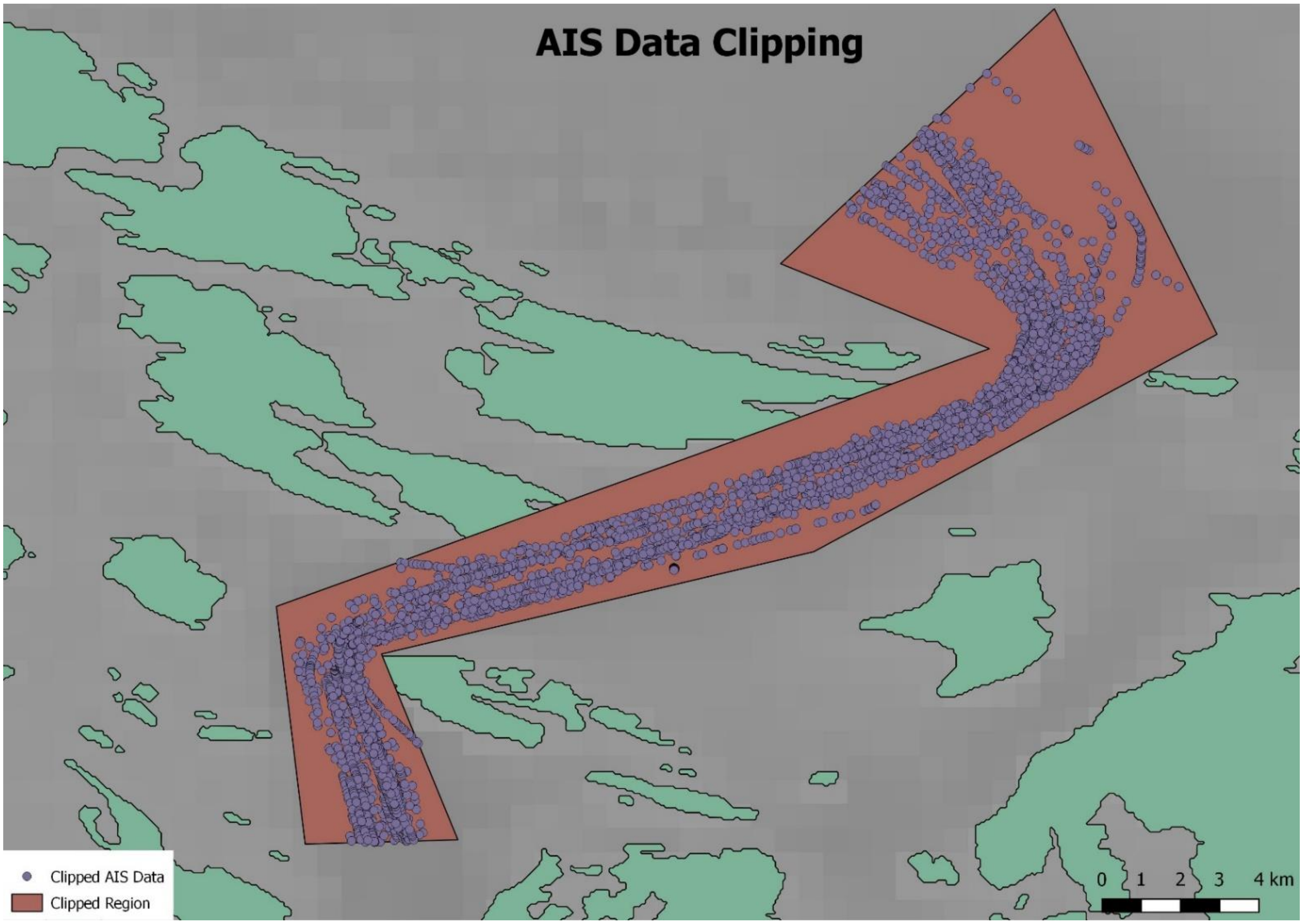


Figure 2.10 Data clipping of raw AIS data (before manual removal of erroneous points)

2.6.2. Heatmaps

With the abundance of point data in the subsample, a heat map was chosen to best quantify different iterations of the model for path likeness (Netek et al., 2018). By creating a heatmap, the model data can be resampled for consistency and using point sampling techniques, given values according to their location on the heatmap. The heatmap function uses Kernel Density Estimation to produce a density raster (heatmap) of a point vector layer (AIS data). The density is calculated based on the number of points in a location, with larger numbers of clustered points resulting in greater values. Heatmaps allow for easy identification of clustered points and provide values that can be used for quantifying the accuracy of each model iteration. When producing the ideal heatmap, there are several parameters that affect the resulting raster image, with the most important being the radius. Radius is used to specify the heatmap search radius (or kernel bandwidth) in map units (meters). The radius specifies the distance around a point at which the influence of the point will be felt. Larger values result in greater smoothing, but smaller values may show finer details and variation in point density. These differences can impact comparison with model results. For this analysis, three different radiuses were chosen: 500, 1000, and 2000 meters (Figures 2.11-2.13 Heatmaps).

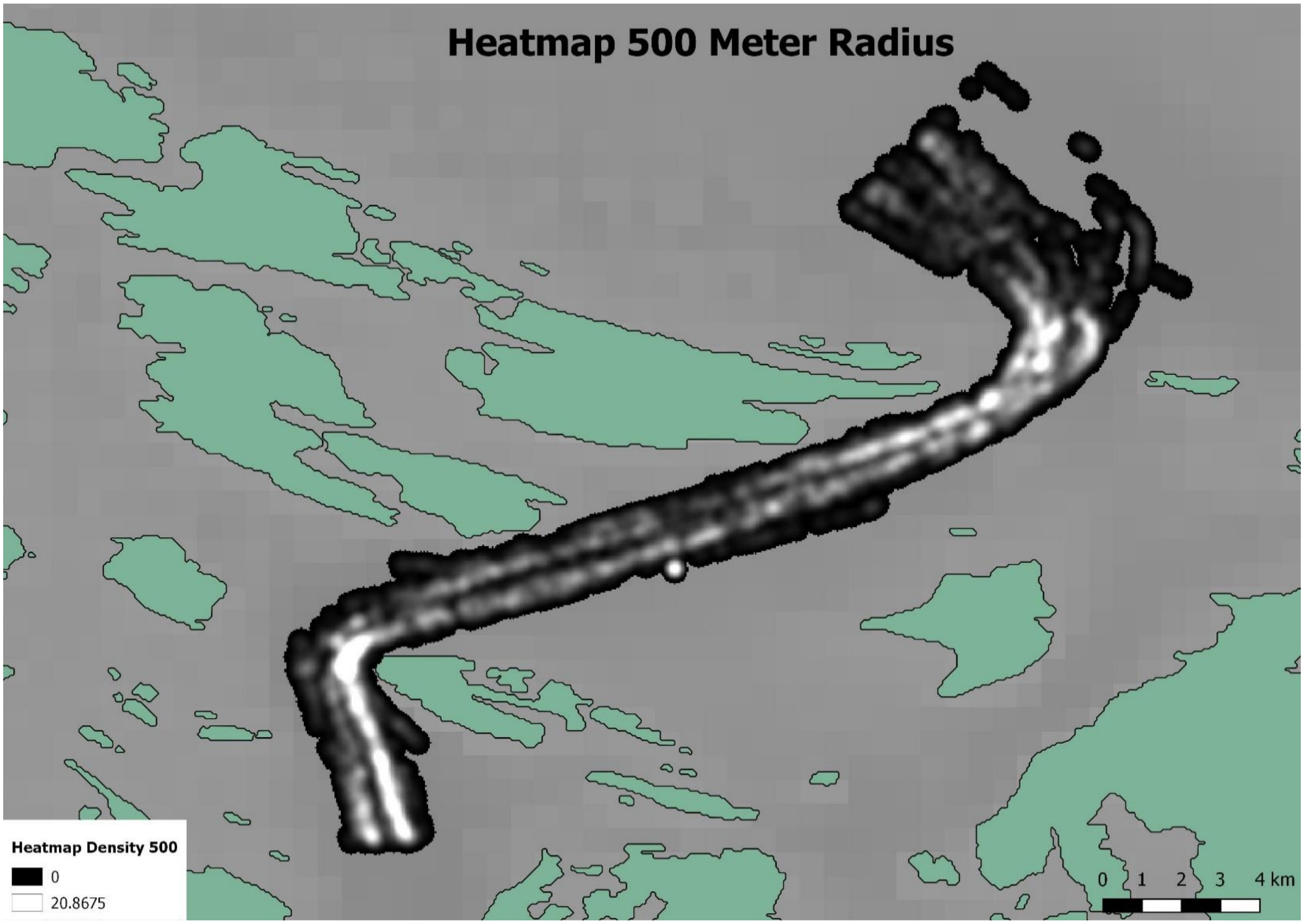


Figure 2.11 Non-Directional heatmap with 500-meter search radius

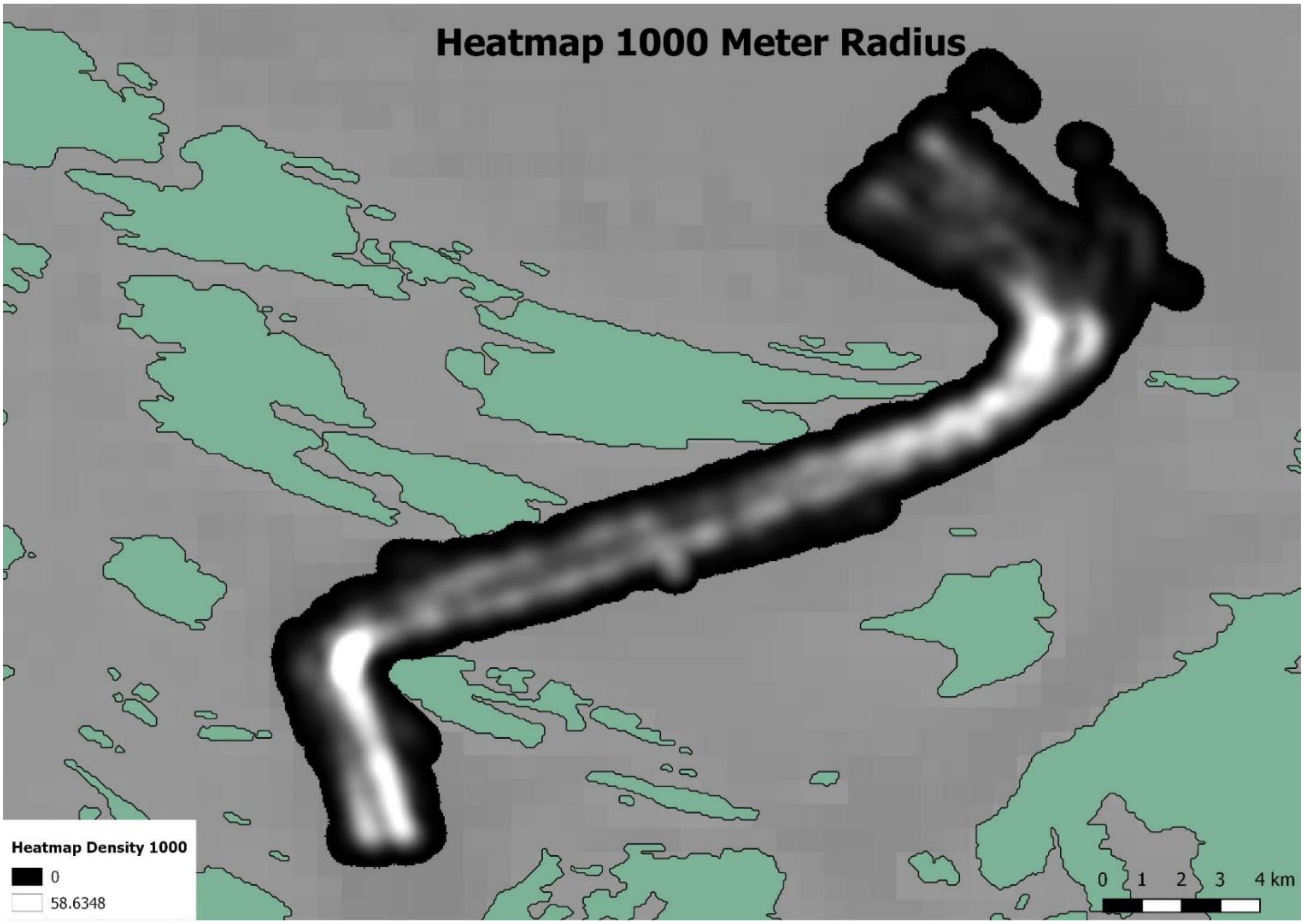


Figure 2.12 Non-Directional heatmap with 1000 meter search radius

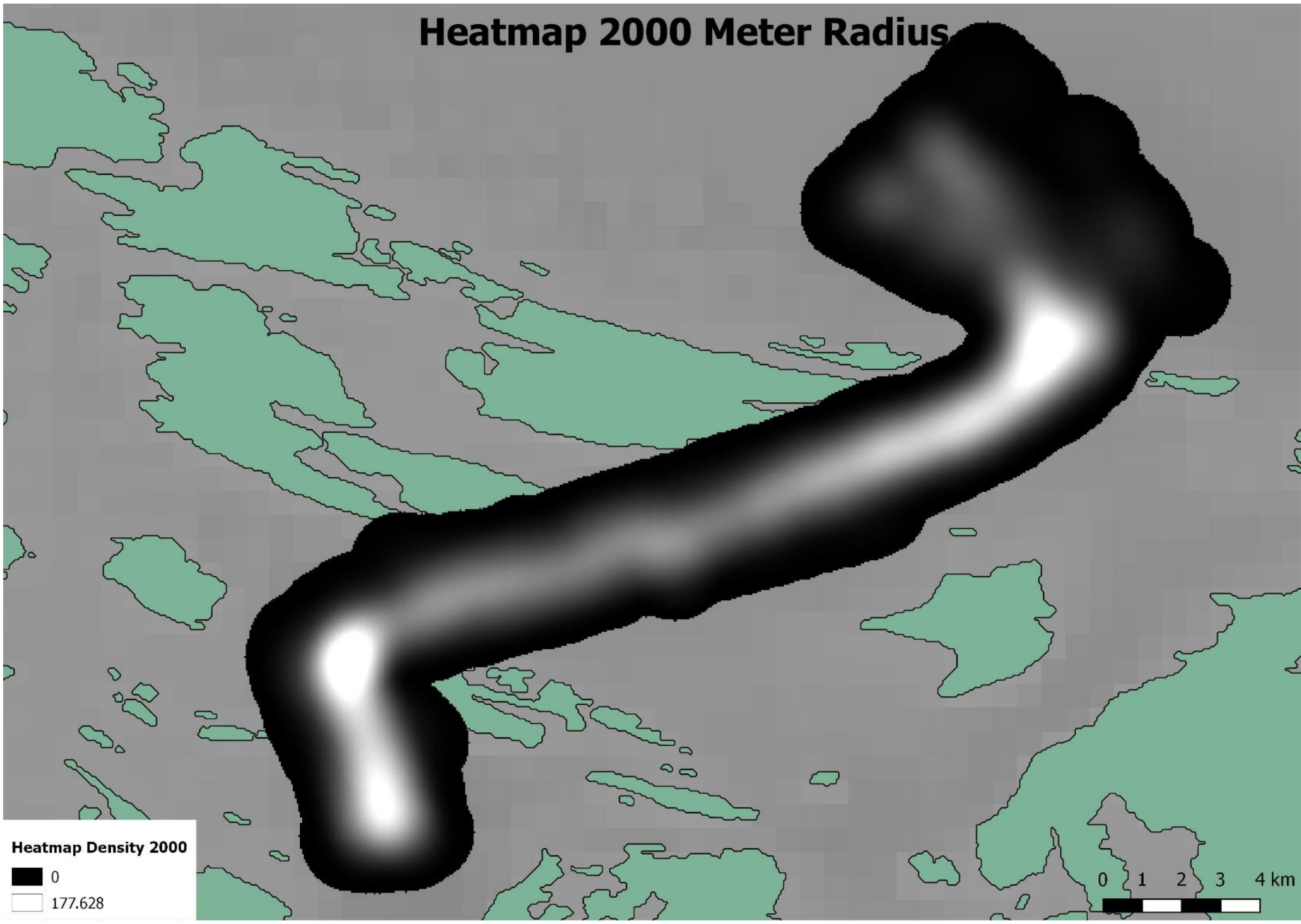


Figure 2.13 Non-Directional heatmap with 2000 meter search radius

The model iterations used for analysis simulate port inbound travel. This differs from the AIS data used for comparison as the dataset includes any ship in the selected location, regardless of direction. To make an accurate comparison, the AIS data was filtered to only include ships travelling towards port. Filtering the data was accomplished by using the heading value to determine the direction of travel for every point in the dataset. All data with an outbound heading was removed from the selection, leaving only ships travelling towards port. This excluded a few erroneous data points that were manually selected and removed from the dataset. The clipping, merging and heatmaps were applied to both directional AIS data and non-directional AIS data.

2.6.3. Resample

To utilize the heatmaps most effectively, all the model data were resampled. This is done using several native and third party QGIS plugins. All simulations have the same velocity or speed (10kn); however, due to slight changes in the path finding algorithm, not all iterations have the same number of points in a given space. Resampling is also necessary to address gaps between model point vector data. The nodes exported from the model are sampled less frequently and create gaps in the heatmap where significant information may be neglected (Figure 2.14 Heatmap Model Data Gaps).

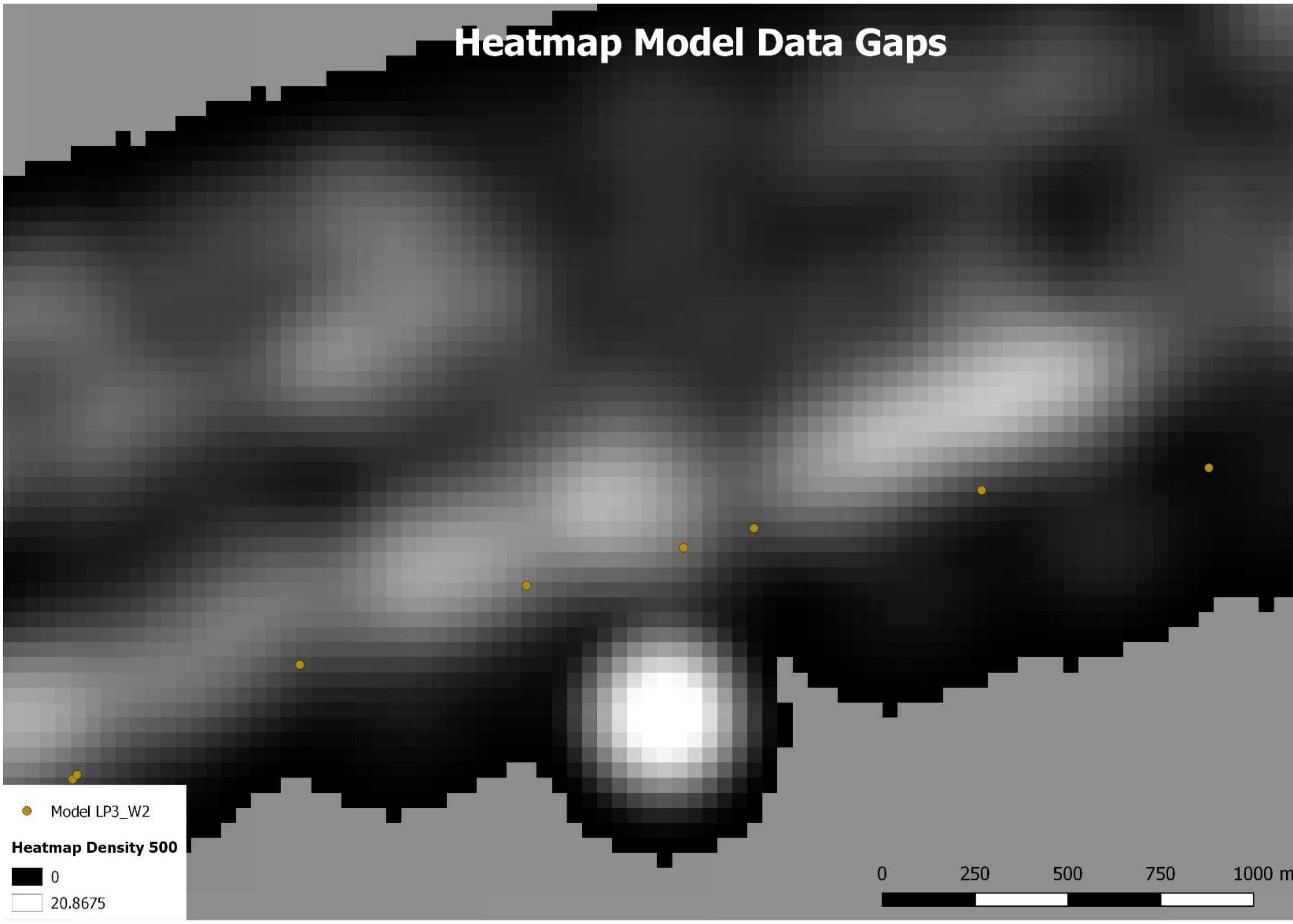


Figure 2.14 Heatmap model data gaps. Emphasizing the importance of resampling as gaps in the model data can skew heatmap results.

The first step in resampling is to create a vector line feature out of the point data. Using the Points2one plugin, the input vector point layer (model iterations) can be output to a line, connecting all the data in a user defined order (sequential) (Figure 2.15 Vector Line Model Data).

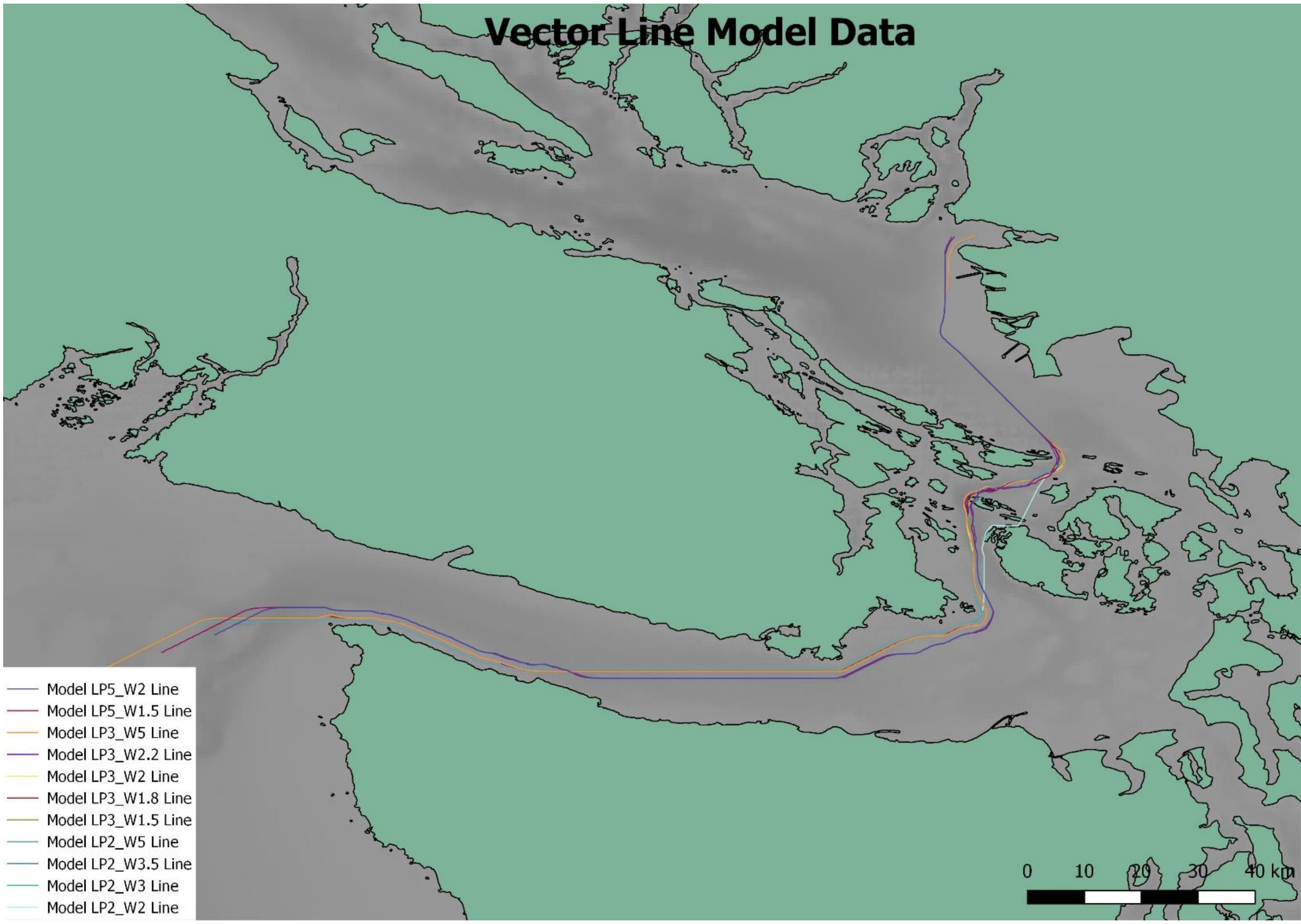


Figure 2.15 Vector line model data. Changing original data output to line data before resampling.

To resample the vector line features back to point data, the Qchainage plugin was utilized. Qchainage operates by creating point data at a given interval along a polygon or line. The interval was set to 0.0005 degrees for a more detailed output without an overabundance of redundant points (Figure 2.16 Resampled Model Data).

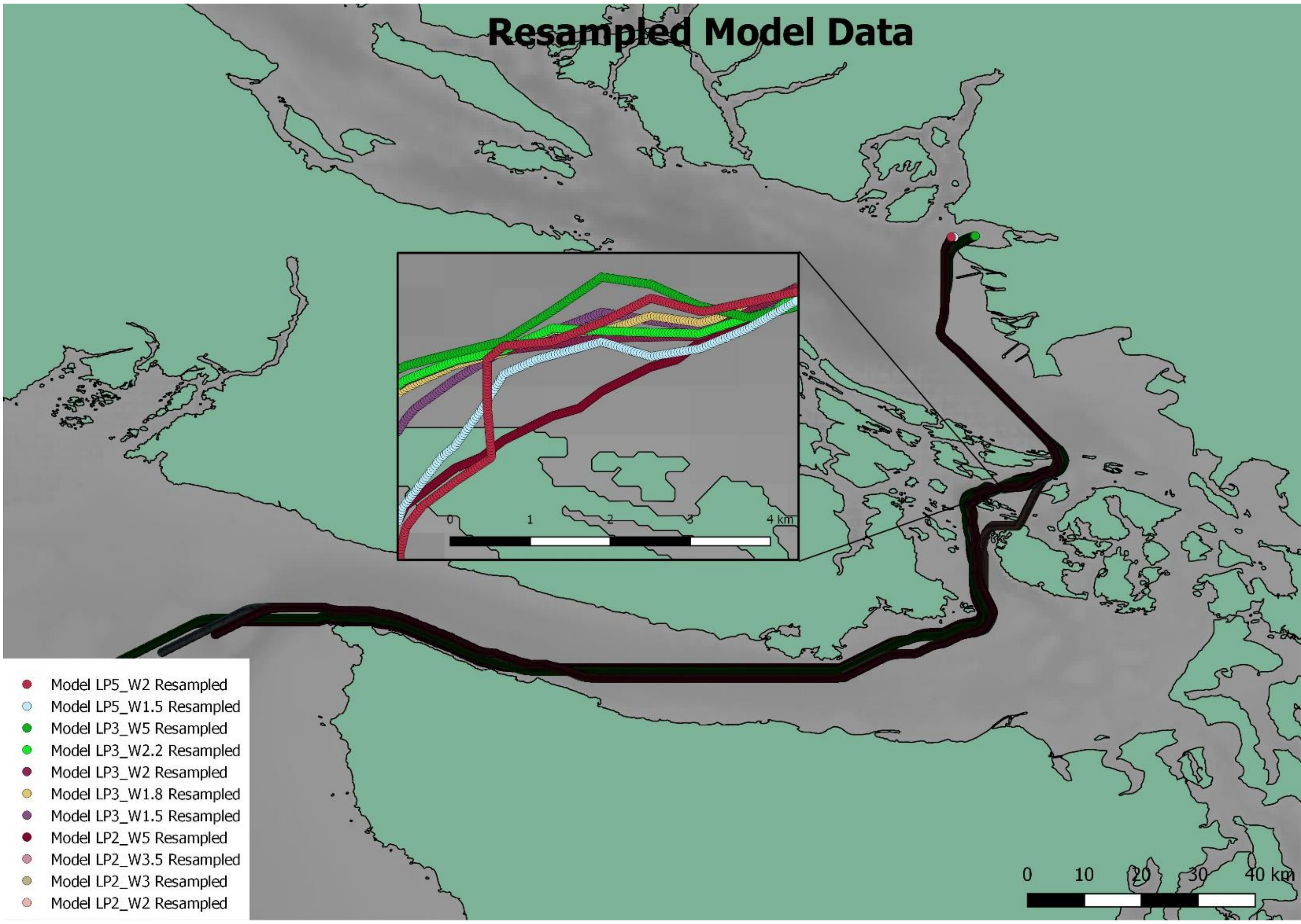


Figure 2.16 Resampled Model Data. Final step of resampling process, evenly distributing model data to avoid slight inconsistencies in each iteration.

With the resampled model data, a more informed comparison with AIS heatmap results can be performed. Using the point sampling plugin collects raster values from a given layer (heatmaps) at the specified sample points (resampled model data). The output creates a new point layer with existing locations taken from the underlying raster image. The new data contains only one column representing its corresponding heatmap value. The metadata was lost when the Points2one function was used, as the vector line feature cannot contain point metadata. The results are used for comparison with AIS heatmaps for path accuracy, and the metadata is no longer needed. These results are then saved as .csv files and opened in Excel for further analysis and visualization.

2.6.4. Ship-Wise Errors

Ship-Wise Errors (SWE) manually select several individual ship point data to determine how model results compare to the selection and how well this type of analysis can inform model route replication (Figure 2.8 Selected AIS data). 6 individual ships (using the MMSI) on a single inbound trip were selected with emphasis on selecting a variety that reduced gaps in the data. The selection was merged and clipped before performing the NNJoin function in QGIS. The NNJoin function uses two vector layers to calculate the distance between the two layers. The results were output as .csv files and exported to Excel for further analysis.

3. Results

3.1. Preliminary Assessment

The MSRM requires human input to find the correct parameters for different waterways. As each location for simulation is unique, the ideal parameters for LP and LPW require adjustment to provide simulation results that resemble existing shipping pathways. A preliminary assessment of the waterway is required to make an informed guess as to what range of values are most suitable for the study area.

For LP, the assessment is based on the width of the narrowest channel in the waterway, amount of variation in channel width and the scale of the map. The evaluation of the waterway is performed by applying a rough starting LP and running the simulation to visualize how many cells the waterway are colored red, indicating that they fall within the LP range. The starting LP will depend on the width of the narrowest channel along the desired path. At this point using the full AIS dataset is not necessary as a rough idea of shipping pathways is sufficient to get a starting point for the parameters in the simulation. Once the route has been identified the LP should ideally be at most, half the number of cells form border to border in the narrowest channel of the route. If the LP is greater than half the number of cells in a given waterway the simulation may select an alternate path. This is dependent on the LPW and the specific waterway in question. If no other route provides a shorter path to the destination (accounting for LP modifications), the algorithm can select a waterway where all cells are within the LP range. In the case of Port Metro, the Haro Strait and waterway around Stuart Island contains the most obstacles and therefore provides the most complicated path finding scenario with the narrowest channel width. Once the LP value is chosen by visual assessment, multiple test runs are performed to help narrow down the ideal LP selection. This is accomplished with visual analysis of AIS data and comparing it to the test model iterations. The same process is applied to determine an ideal range of LPW values.

After the initial assessment of the MSRM three different LP values were chosen. Each land proximity value was chosen to illustrate important aspects of the MSRM. Due to the variety in the waterway, many different LPs were tested until the best values for analysis were identified.

LP3 displays how the model can function when the land proximity is appropriate for the entirety of the given waterway. With an appropriate LP, the LPW can be adjusted by smaller increments and provide more variation in path selection. As a result, five LPWs values were selected to show the variety in path selection - more than any other LP used for analysis. LP4 was also experimented with but provided similar results to that of LP3. It was not included in the analysis as having a variety of model iterations examining model functionality in all cases, especially at its limits, is of more importance.

LP2 was chosen because it shows how the model functions with minimal adjustment to the default BFS algorithm. A land proximity of 2 does not alter many cells in the waterway creating paths most similar to an unmodified BFS algorithm. This is most noticeable with a low weight, for example LP2 W2. With these parameters the model would select a completely different path when navigating the Haro Strait past Stuart Island (Figure 2.5 Raw model data). As a result, the LPW would have to alter the original BFS values by a large enough sum to choose a path that avoids areas near land.

LP5 illustrates how the model performs when the algorithm modifies too many cells for the width of the given waterway. When the LP is set above 4 the model begins to produce inaccurate route replication. This is due to the narrow width of many of the waterways through the study area. Because the distance in some areas is less than 10 cells from island to island, an LP of 5 or higher creates pathways where every cell is multiplied by the LPW.

The same process of trial and error was applied to identify the ideal LPW range for each LP value. Once the LP is chosen the LPW is experimented with to see what value creates the most accurate path. Careful selection of LPWs were chosen to best analyze the study area. When adjusting the LPW, small deviations will not change the BFS values enough to alter route choice. A variety of LPW's are examined in analysis to see how changes in LPW effect model route replication.

When importing the MSRM data from QGIS to Excel, several data clean up procedures are necessary. Data cleaning is needed to address slight differences in the start and end points of model iterations and to adjust model results with null values for points that fail to occupy the area of the heatmap radius. Excel was used to modify all the null values in the data series to 0's and to align the model iterations so they share the same start and end cells.

3.2. Model Results

The results compare simulation iterations in the following ways: average heatmap scores, sequential point data compared to heatmap scores, ship-wise errors, and directional AIS data vs non directional AIS data. All the data collected in QGIS were imported to Excel for further analysis and data presentation.

The average heatmap scores can help determine which model iterations perform better for the entirety of the simulation length. Using a different heatmap search radius allows for more flexibility when analyzing the model iterations and illustrates the change varying radii has on the model results. Sequential point data from model iteration in relation to heatmap scores is important for identifying location specific model scores. Examining average heatmap scores does not provide an adequate representation of model performance however, in tandem with sequential point data results, it is possible to identify where the model performs better and how it impacts average scores. Without the need for visual assessment of model iterations, the average scores and sequential results can provide a basis for analyzing model iteration performance.

Ship-wise errors use a small subset of data to analyze model iteration performance. This can help show how individual ship journeys compare to model results however, due to the nature of AIS data, gaps and insufficient location data hinder effectiveness. Using heat maps is crucial for accurate analysis and helps emphasize the benefits of using large data sets for comparison. Including directional and non-directional results highlight the need for data cleaning when using large data sets and when comparing the two, provide more information on model iteration route selection.

3.2.1. Average Heatmap Scores

Heatmap scores were derived for all model iteration points within the heatmap range. The heatmap value for the cell that the MSRM point data is occupying is sampled using the 'sample from raster data' function in QGIS. Point data from model iterations are given a new attribute field that corresponds with each of the heatmap radii (500m, 1000m, 2000m). These values are considered heatmap scores and used for analyzing how well model iterations replicate the AIS data. For Average heatmap scores, the average for each model iteration and heatmap radii is graphed using excel.

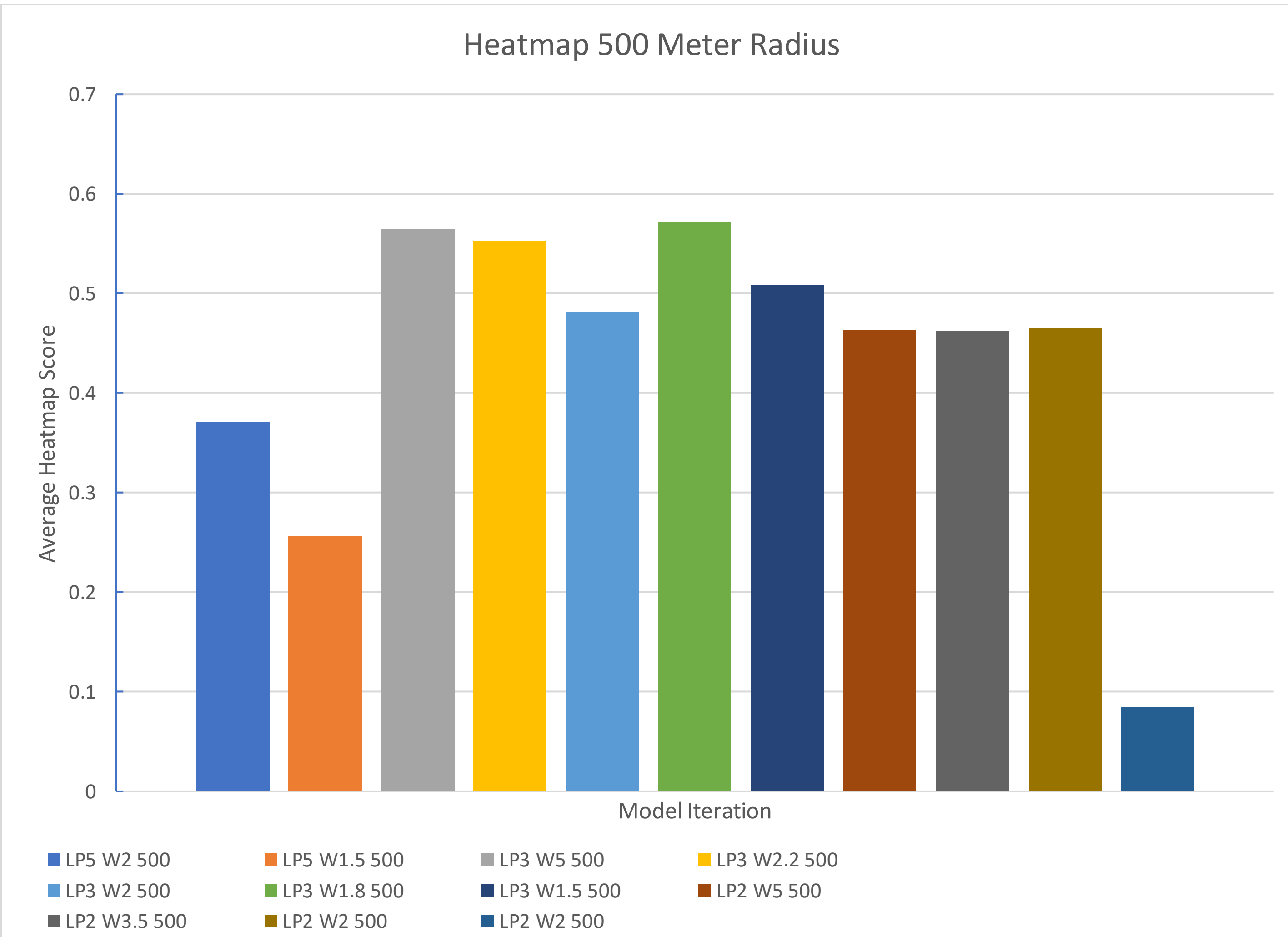


Figure 3.1.1 Directional Heatmap Results 500 Meter Radius

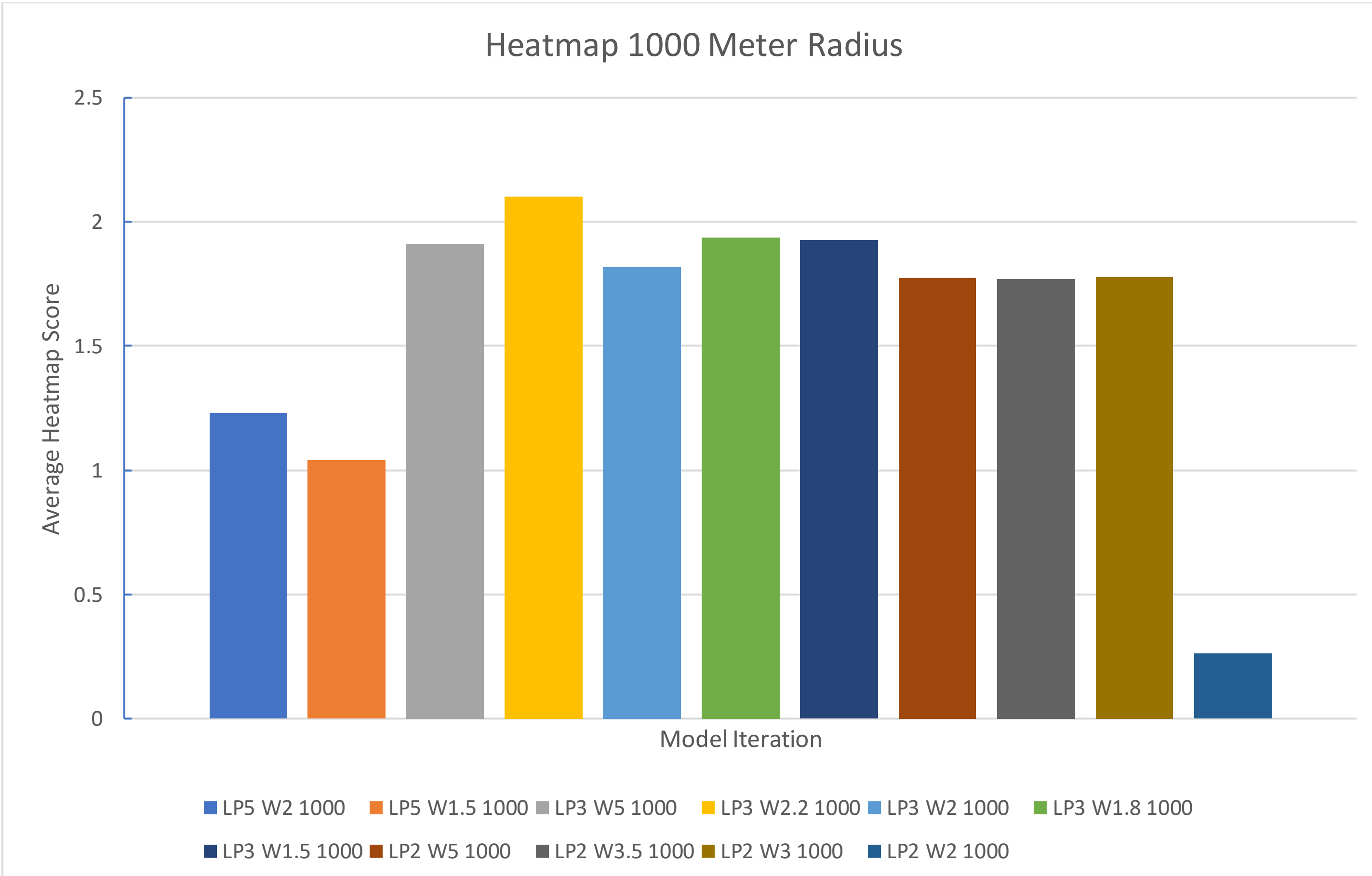


Figure 3.1.2 Directional Heatmap Results 1000 Meter Radius

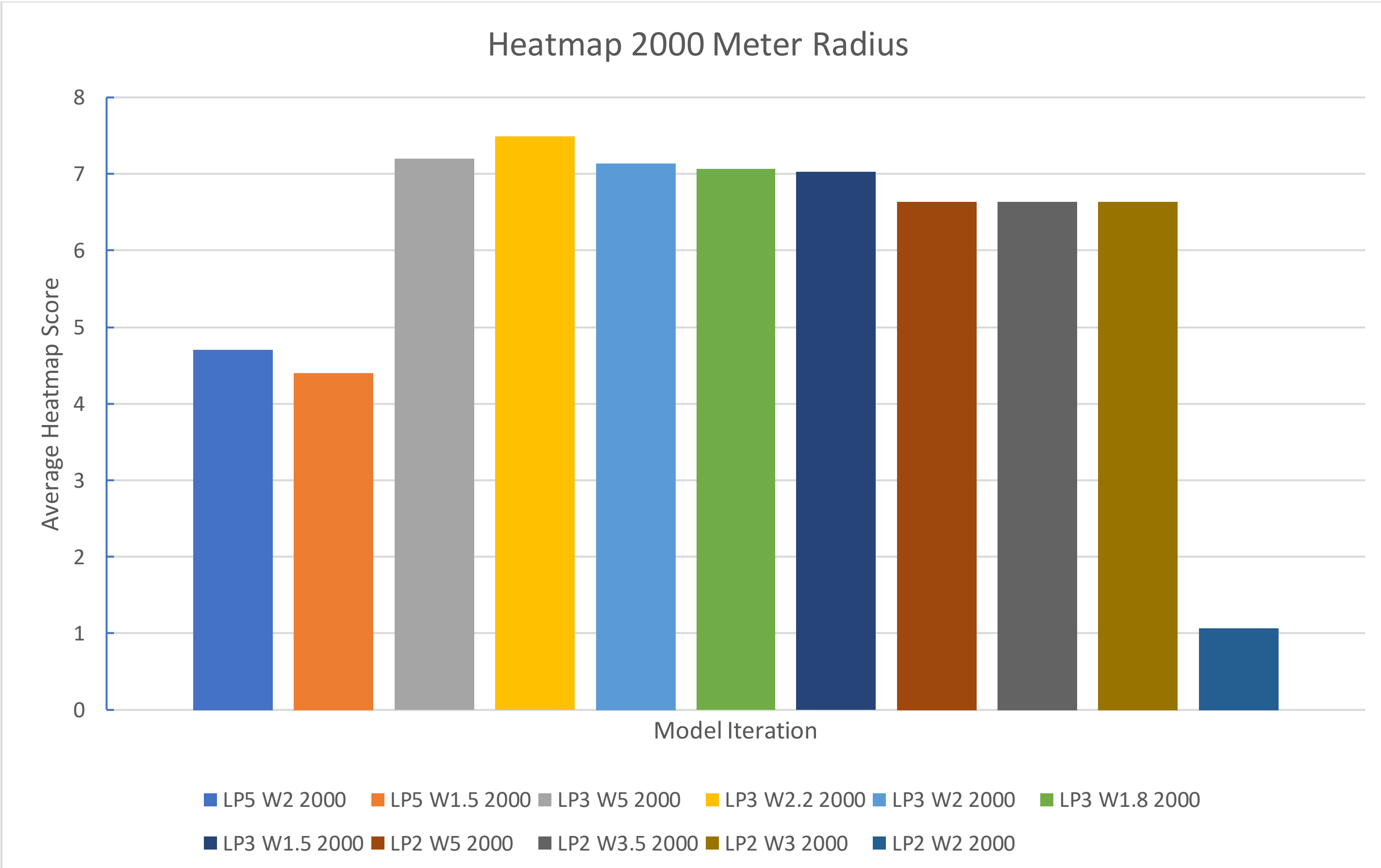


Figure 3.1.3 Directional Heatmap Results 2000 Meter Radius

3.2.2. Heatmap Radius

The model iterations and the various heatmap radii are both considered when analyzing the results. Before looking at the performance of individual model iterations, the changes in heatmap radius must be addressed. There are several reasons different heatmap radii are important to consider, especially when using a database with thousands of points. With big data there needs to be considerable data clean up to most effectively analyze the results (Netek et al., 2018). The data used for analysis is directional to match the model output and ships that may be anchored are identified using their SOG (Speed Over Ground) and removed. Furthermore, visual identification of erroneous data points was necessary to remove data points that are accidental with ships travelling through the study location but on route to a different destination. Even with the most stringent data clean up there may still be erroneous data points that exist without possible identification (Gunnar Aarsæther & Moan, 2009). Using heat maps can help reduce the impact of data errors when analyzing large-scale datasets, as it will reduce the impact unusual ship routes will have on the results.

The nature of maritime travel and AIS data accuracy creates slight inconsistencies between the actual path ships take opposed to the predefined route set by Port authority. Heatmaps use cluster analysis to apply higher values to areas with an abundance of points to signify where ships most frequently travel. This can also create issues as AIS data collected from static ships will have many points overlapping giving it very high heat map scores. To avoid this, all ships with a speed of 0 were removed from the data set. 3 heatmap radii were carefully chosen for analyzing model results. 500 m (figure 3.1.1), 1000 m (figure 3.1.2) and 2000 m (figure 3.1.3) were selected to show how applying different radii will affect the results.

With a smaller heatmap radius, any model iteration points that fall outside the search radius are given a null value regardless of their proximity to the heatmap range. This provides another reason to analyze multiple heatmap search radii as some model results may only slightly fall outside the heatmap radius but will receive the same score as model points that are much farther from the heatmap radius.

The different heat map radii emphasize two major observations. First is that the changes in the average heatmap score is more variable the smaller the radius is. The smaller the radius, the more precise the model iteration scores will be, slight deviations from the AIS data will change average scores more drastically. By examining the average scores for 2000m radius the variation in the average heatmap scores is reduced. The average scores for 2000m radius heatmap are within a range 0.5 for LP3(high score of 7.48 and a low of 7.02), where the average scores for 500 m radius heatmap have a range greater than 0.8 (high score of 0.571, low of 0.481). The greatest amount of variation in the 500m radius map is due to LP3 W2 scoring particularly low in comparison to the other model iterations with a LP of 3. By comparing the sequential data for LP3, when the radius for the heatmap is set to 500m, LP3 W2 has low heat map scores for the beginning of the simulation route. When examining the raw model data, it is evident that LP3 W2 fails to match the AIS data with the same level of accuracy as other model iterations with LP3. When the radius is increased, the amount of variation is reduced as slight routing discrepancies around Stuart Island are not as impactful on average heatmap scores. LP3 W2 continues to perform better as the heatmap radius is increased.

The second observation is that LP order from best performing to worst is the same across all Heatmap radii. The order of LP performance is expected as the different LP values were selected to illustrate model functionality. The LPW however, changes quite considerably between the different heatmap radii.

When looking at the results of the average heatmap scores, the differences that are seen in the radius changes are not conclusive on their own. Due to the nature of heatmap comparison, average scores are not always indicative of the best overall model performance in terms of route replication. The model path may be very accurate for a small portion of the route which will greatly increase the scores for a certain area. This can be seen in LP3 W5 and will be addressed further in the following section. This is largely mitigated by resampling with a small enough distance between points to reduce skewing the average results due to single points with very high scores. Although the high scores in certain areas may indicate the best route selection for the specific area in question, when looking at model performance, consistent route replication for the entire heatmap range can be considered more valuable depending on how you are looking at the results. By having larger radii, the impact of data outliers can be mitigated by increasing the effective score range of large data clusters. This increases the scores for model iterations that may have received lower scores with a smaller heatmap radius. Visually analyzing the model results and heatmaps in GIS software can help identify model iteration performance at different points of the simulation. Although this can provide enough information to help identify strong and weak points in model iterations, having a numerical representation can greatly benefit analysis. By using sequential point data from model iterations in comparison to average heatmap scores, it becomes possible to determine model performance at different points in the simulation.

3.2.3. Sequential Point Data Heatmap Comparison (SPDHC) and Model Iteration Performance

SPDHC uses the same heatmap scores calculated with the sample from raster data function, however it only includes heatmap analysis of the 500m radius. The extent of the heatmap includes model iteration point data from an approximate range of 4500 to 5500 in the data series. The SPDHC graph for each LP shows where the model has higher scores and is therefore in closer proximity to hot spots in the AIS data heatmap.

SPDHC is useful for comparing how model iterations perform at different locations in the study area. Any change to LP or LPW will alter the route generated for each waterway scenario. Examining performance at different points in the waterway is crucial as it can indicate how well the model replicates AIS data at different stages of the simulation. When looking at maps with simulation results and heatmap data, it can be difficult to determine heatmap scores. The SPDHC can assist in locating exact scores throughout the simulation area. This information can help determine how location specific performance affects the average heatmap scores. The SPDHC can help provide information that average heatmap scores cannot. By using SPDHC, detailed analysis of each model iteration can be performed. This section will focus on the different heatmap iteration performance. The preliminary assessment provided a basis for analyzing how each model iteration performed compared to predetermined maritime shipping pathways. Without comparing the results to a large AIS dataset, it is not immediately evident how the model iterations perform.

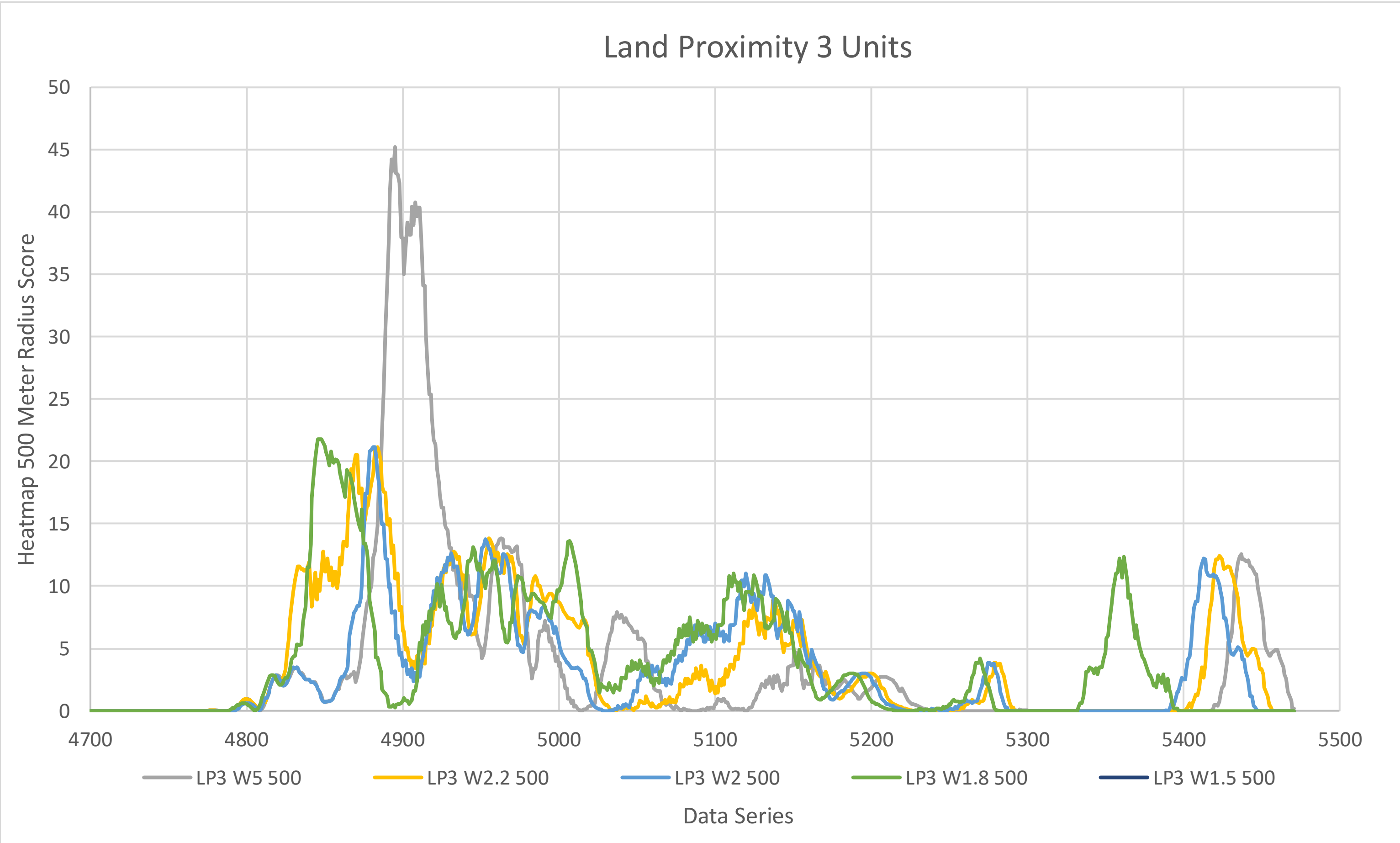


Figure 3.2.1 Sequential Data 500M Heatmap Directional Land Proximity 3

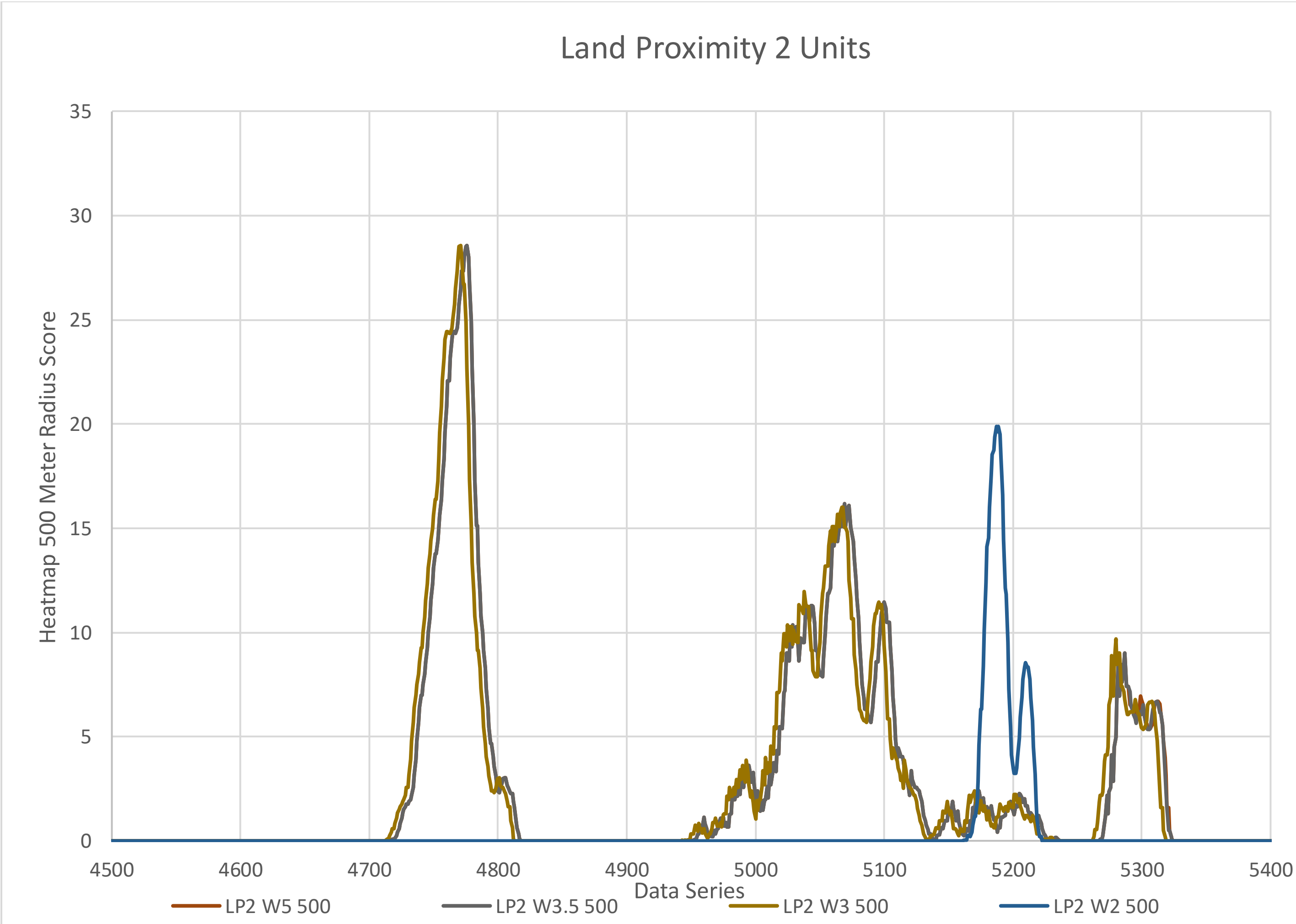


Figure 3.2.2 Sequential Data 500M Heatmap Directional Land Proximity 2

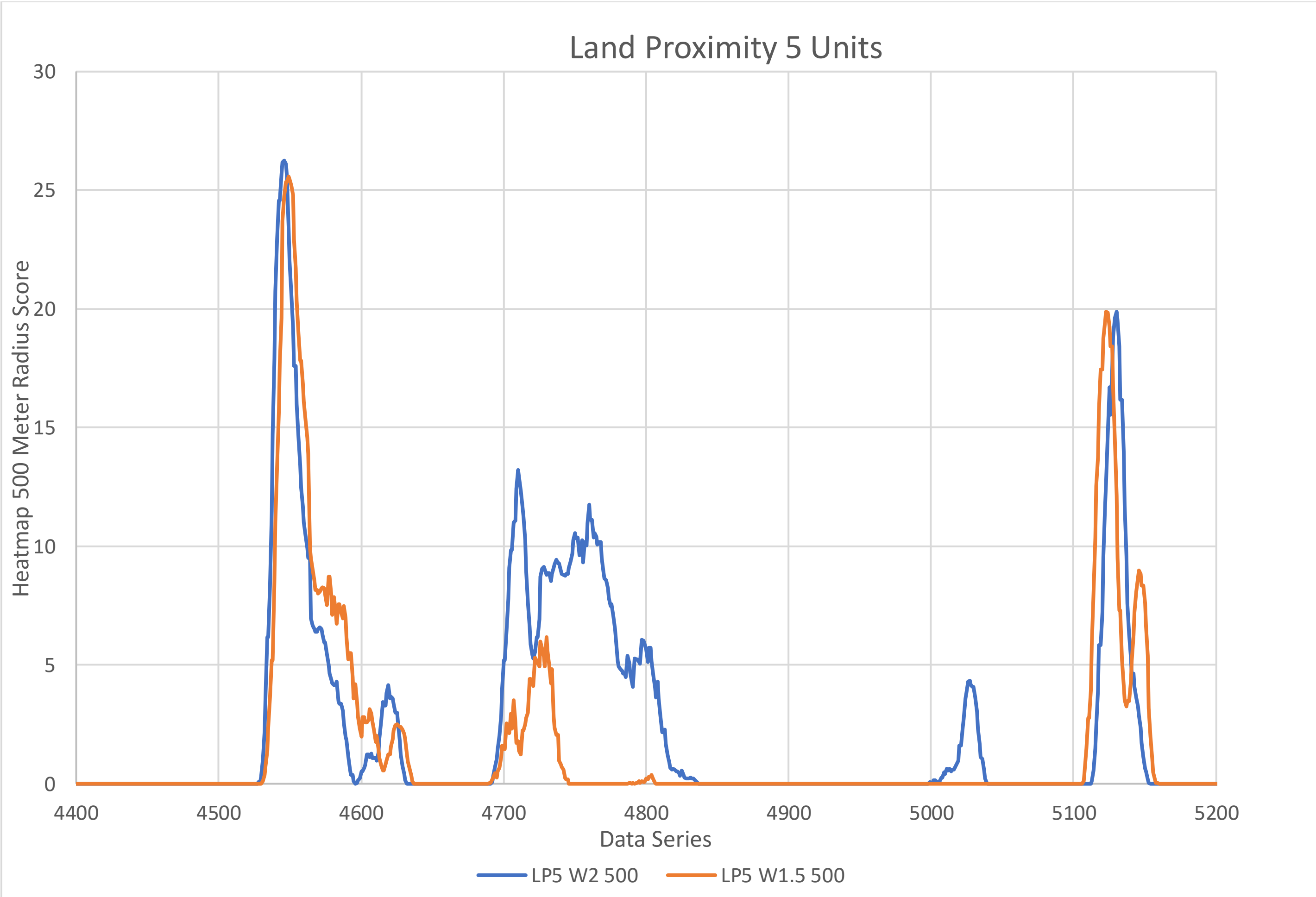


Figure 3.2.3 Sequential Data 500M Heatmap Directional Land Proximity 5

3.3. LP3

Upon analyzing the average heatmap results, LP3 performed the best followed by LP2 and LP5. The SPDHC demonstrates further the performance characteristics of each LP. As mentioned in the heatmap section, LP3 ranks the highest regardless of the heatmap radius. Analyzing the SPDHC for LP3 shows consistent scoring throughout the simulation. The other LP examples fail to receive scores for much of the simulation length. However, high scores for small sections and relatively low scores in the same area for LP3 examples, allow average heatmap scores to remain close. Many of the spikes in scores for model data exist at the waterway around Stuart Island. Due to the collection protocols for AIS data, more point data is collected when ships are turning - as mentioned in the AIS Data section. This creates clusters of AIS data which skew average heatmap scores. SPDHC comparison provides insight into overall model performance by identifying data spikes. Visual analysis of model iterations shows that LP3 is the most consistent for the entirety of the simulation as it navigates the Stuart Island turn more accurately than the other LPs.

The average heatmap scores at 2000m radius for LP3 provide insight into model functionality. When the heatmap radius is increased, it reduces the impact of data spikes where model iterations receive high scores for one section of the model. This allows average scores to be more indicative of overall route replication when compared to lower heatmap radii. These results also back up what can be visually seen when analyzing the model results. The average heatmap scores for 2000m show that the increase in LPW provides better overall route replication until the weight is higher than LP3 W2.2 - where model performance begins to degrade. A limit exists as too much obstacle avoidance will create pathways that encroach on incoming traffic lanes. Non-directional results support these findings as LP3 W5 scores the highest amongst all model iterations for the non-directional heatmap examples. Due to LP3 W5 having the highest weight, it selects a path that at times occupies opposing traffic lanes.

LP3 W5 performs better at directional analysis when the heatmap radii is reduced. This is mainly due to other LP3 models receiving lower scores around Stuart Island, where a large cluster of AIS data points exist. LP3 W5 scores highest at one small section, which is emphasized with a smaller radius. When looking at the SPDHC (Figure 3.2.1), LP3 W5 receives maximum scores more than double any other model with LP3 (21 to 45). This allows LP3W5 to seemingly perform better with 500m radius when looking at average heatmap scores on their own. Once the corner around Stuart Island is navigated, having a LPW of 5 negatively affects the heatmap scores. The South Pender Islands on route to the Strait of Georgia create a buffer that pushes the route outside of the 500m heatmap. This is evident when looking at the SPDHC graph for LP3, all model iterations except LP3W5 receive scores at data series 5100-5200.

LP3W2 is an outlier as it performs worse than other LP3 models with lower weight. When examining model results, LP3 W2 occupies a wider turn at the end of the Haro Strait approaching Stuart Island, explaining the lower average heatmap score. Analyzing the SPDHC for LP3, scores at data series 4800-4900 are responsible for the performance difference. This is like LP3W5 however, LP3W5 scores much higher due to having a large spike in scores when it navigates the Stuart Island turn. As a result, LP3W2 scores much lower than the other LP3 model iterations with 500m and 1000m heatmap radii.

When examining model iterations with LP3, the changes in LPW effectively increase the amount of obstacle avoidance - in this case Stuart Island. As the LPW is increased the model iterations take a wider turn around the island. This is even more evident when looking at model iterations with LP2. LP2 takes a more direct path cutting through Stuart Island (due to the scale of the bathymetry map the model identifies these cells as reachable). This proves correct model functionality as an increase in obstacle avoidance is expected as the weight and land proximity are increased. A limit to the amount of obstacle avoidance (LP value) exists in waterways with more than one obstacle. With only one obstacle, the LP can be as high as the map limits allow. Large modifications to BFS values with high LPW's will not create errors as the simulation will be able to find a route outside the range of affected cells. If the study area has multiple obstacles the affective LP zones can overlap causing errors when trying to choose a path. LP5 is a good example as the LPW cannot be much higher than 2 to avoid any errors from occurring. Figure 3.2.4 (LP5 path finding algorithm example) shows that much of the waterway in the study is colored red and therefore falls under the influence of the LP. More analysis of LP5 will be addressed in the LP5 section.

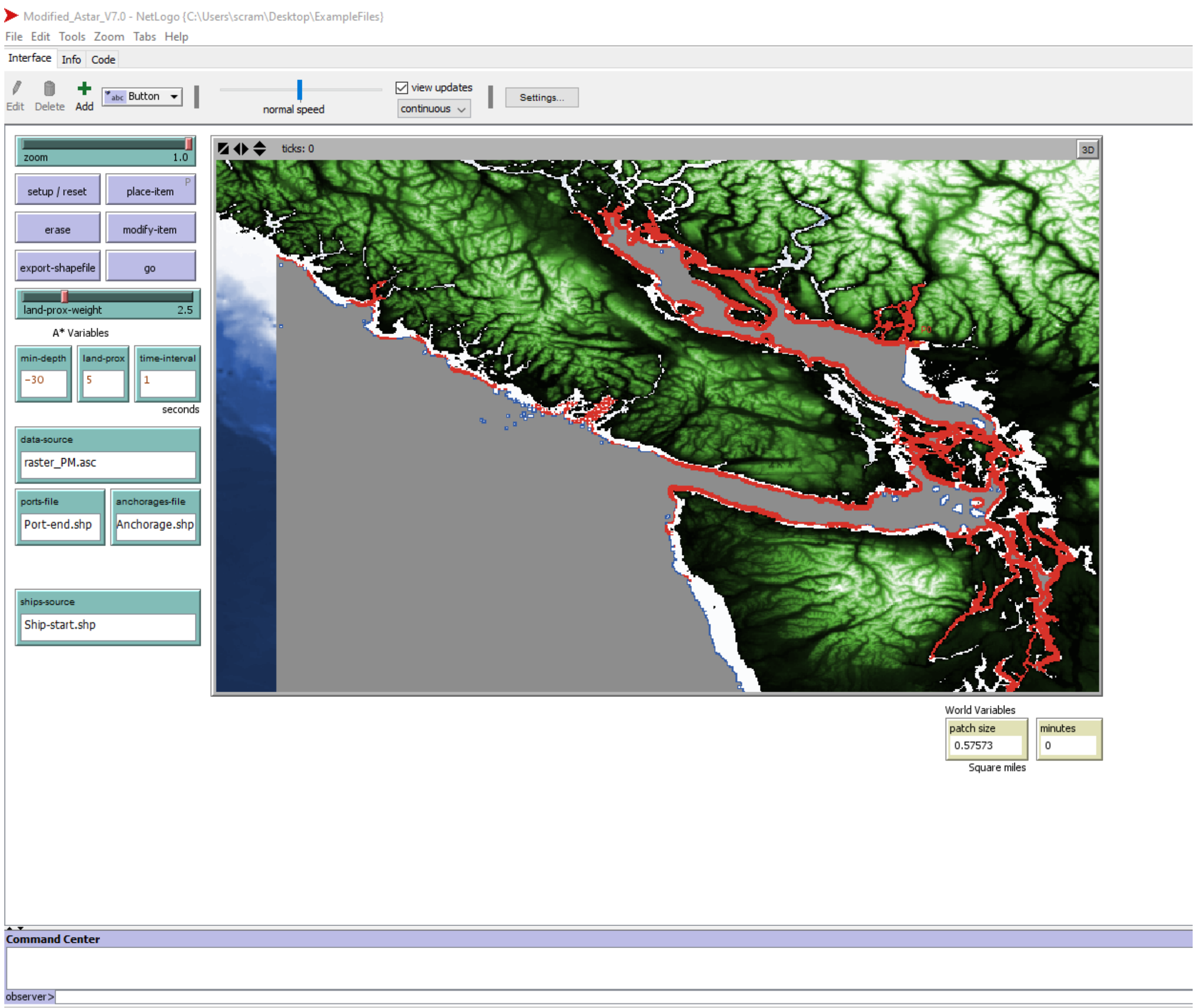


Figure 3.2.4 LP5 path finding algorithm example. The number of red pixels indicate how many nodes in the waterway are being affected by the LP distance.

The effects of changing the LPW are most noticeable with LP3. For instance, the order of performance from best to worst at 500m is LP3W1.8, LP3W5, LP3W2.2, LP3W1.5, LP3W2. For

1000m the model performance order is LP3W2.2, LP3W1.8, LP3W1.5, LP3W5, LP3W2. Finally, for 2000m the model performance order is LP3W2.2, LP3W5, LP3W2, LP3W1.8, LP3W1.5. Model iterations with LP2 and LP5 have less variation when changing the heatmap radius. Accurate choice of LP for a given waterway can be indicated by an increased variation in scoring. Altering heatmap radius causes model points that are within proximity to AIS data to experience the most variation in scoring. When the route chosen selects a path more like AIS data, changes in the heatmap radius will vary scores more significantly than model iterations that fail to replicate routes as effectively.

3.4. LP2

LP2 performs second best and experiences the least amount of score variation as the heat map radii is increased. However, LP2 has the largest score discrepancy due to LP2 W2. LP2 W2 is a good example of what happens if the standard BFS algorithm is not modified. With these parameters the model selects a completely different path when navigating the Haro Strait region (Figure 2.5 Raw Model Data). With an LP of 2, the LPW needs to be large enough to select a path that is possible for large vessel transit. The shortest path with an unmodified BFS algorithm is unlikely to take the same route as maritime vessels. The most obvious reason is that the model lacks any traffic rules or safety protocols where, ships could not safely travel the shortest path to their destination. Furthermore, inaccurate basemap values allow pixels above land to appear safe for travel but, maritime vessels are unable to travel the same route. If the scale of the raster image is too large, the average depth of a pixel may appear lower than sea level however in reality a large ship may ground.

As the LPW is increased for LP2, the model creates pathways more like model iterations with LP3 and as a result has increased heatmap scores for the entirety of the simulation area. This is evident when looking at the results for LP2 W3, 3.5 and 5. These model iterations perform substantially better than LP2W2 and have scores close to model iterations with LP3. Although LP2 fails to accurately navigate the Stuart Island turn, by analyzing the SPDHC for LP2, high scores at the beginning and the end of the simulation improve performance. This is supported with visual analysis of the model results (Figure 2.5 Raw Model Data). LP2 W3, 3.5 and 5 travel through a cluster of AIS data before navigating Stuart Island that LP3 models do not. This slight change in route provides high scores for LP2 increasing average heatmap scores. Towards the end of the simulation there are less obstacles to avoid and path selection is most similar for all model iterations.

Excluding LP2W2, all model iterations with LP2 take the same path for the entirety of the simulation. The scores are separated by less than 0.002 at 500m, 0.01 at 1000m and 0.01 at 2000m. This is due to slight routing changes towards the end of the simulation. This is evident when analyzing the SPDHC for LP2, as small score discrepancies can be seen around data series 5300. For the remainder of the simulation, model results overlap as they take the exact same route. With a low LP, fewer cells in the simulation have altered BFS values. Changes to LPW values must be much larger to see any changes in route selection. Without as many cells in the waterway under the influence of the LPW, route selection remains the same even with different LPW's.

3.5. LP5

LP5 provides insight into model functionality when many of the cells in the waterway are altered by the LPW (Figure 3.1 LP5 path finding algorithm example). Narrow waterways in the study area prohibit LP values higher than 5. Furthermore, the weight for LP5 cannot be higher than 2 to prevent the model from running into errors. This is directly related to the find exit function, explained in the Methods section. The find exit function allows for ships to reach destinations by recalculating BFS values for the remainder of the trip. This only applies to ports or anchorages that are close to land that are impossible to reach in some scenarios - even with a lower LP and LPW. Solutions to this problem are addressed in the discussion section.

Visual analysis of model results depicts route selection like LP2 when navigating the Stuart Island turn. This goes against perceived model functionality as a larger LP would suggest more obstacle avoidance and therefore a wider turn around Stuart Island. Due to the larger LP, other obstacles (islands) near Stuart Island alter BFS values of patches that are not modified with a smaller LP. The route around Stuart Island is no longer favored as all the BFS values in the waterway have been modified. As seen in LP2, the shortest path will select a route that turns through Stuart Island instead of correctly navigating around it. As a result, LP5 selects a path that does not take a wider turn when navigating Stuart Island and instead takes a path that resembles LP2.

Analyzing the SPDHC shows similar scores for LP5W2 and LP5W1.5 at the start and end of the simulation (data series 4500-4600 and 5100). The difference in scores comes from data series 4700-4800 where LP5W2 receives much higher scores providing it with a better average heatmap score. A higher LP creates more variation in BFS values and will result in increasing route selection variation with small changes to LPW. Larger LPs also create more frequent direction change when selecting a path. Both findings can be seen with visual analysis of model iterations with LP5 (Figure 2.5 Raw Model Data).

3.6. Ship wise errors

Several ship trips from the AIS dataset were handpicked to produce a complete route through the study area. By comparing nearest neighbour (NN) values for model iterations, performance characteristics can be assessed (Figure 3.3.0). Nearest neighbour values are determined by measuring the distance between sample point data and the closest point data in each model dataset. The x-axis corresponds to every point in the sample data set and the y-axis is the distance to the closest point for the corresponding model iteration. Examining ship wise errors provides a simplistic analysis of model iterations. These findings are less accurate than heatmap results and do not provide the same level of insight into model performance.

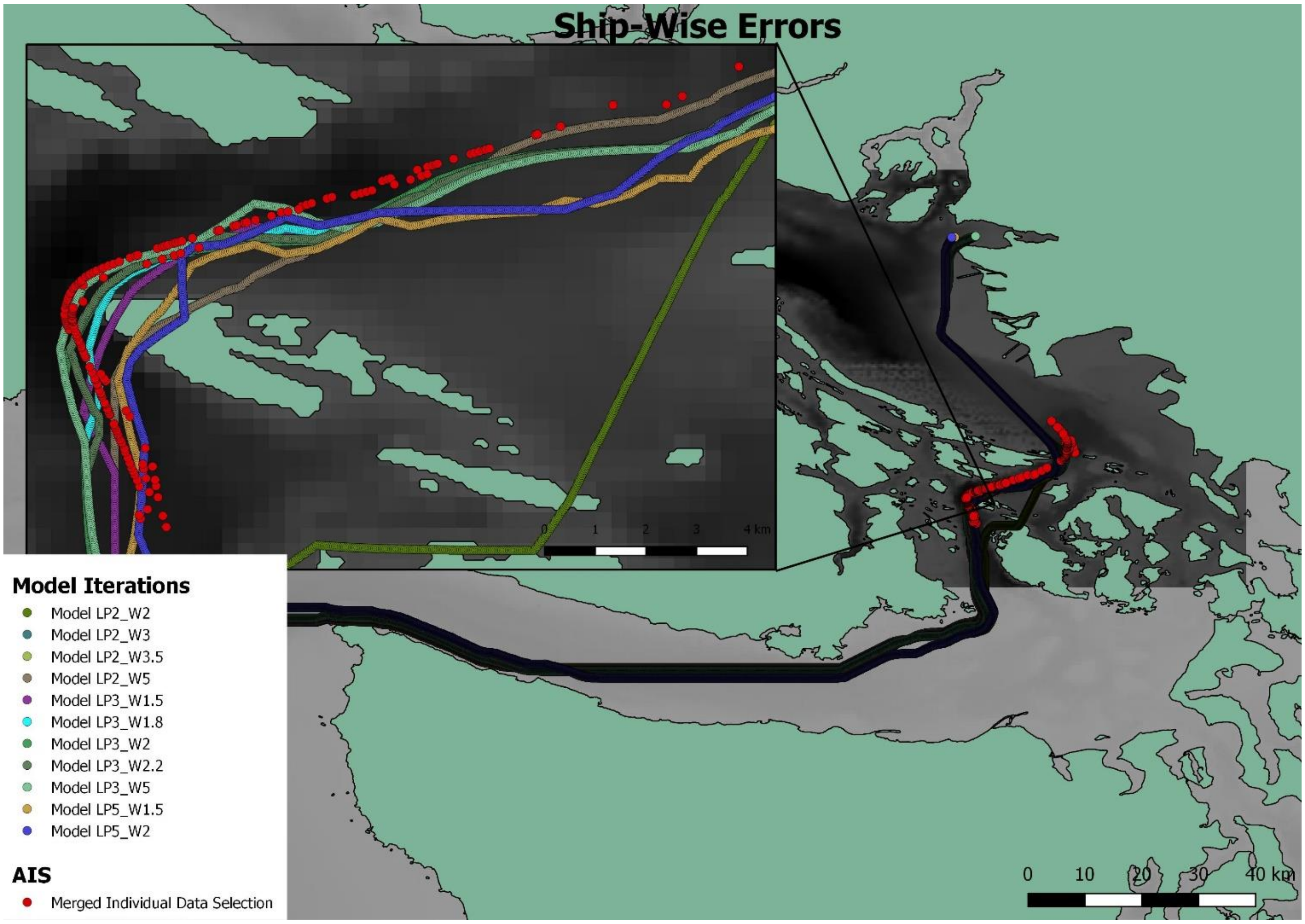


Figure 3.3.0 Ship-Wise Errors and Model Iterations. Provides spatial context

LP2 SWE results provide the most similar findings to heatmap results when compared to other model iterations. This is due to the obvious differences between LP2 model iterations. The large difference in route selection from LP2W2 and other LP2 iterations can be deduced from analyzing the SWE results. Furthermore, the identical path taken by all other LP2 iterations can be seen in SWE results as W5, 3.5 and 3 have the exact same values, appearing as one line in the graph below (LP2 W3).

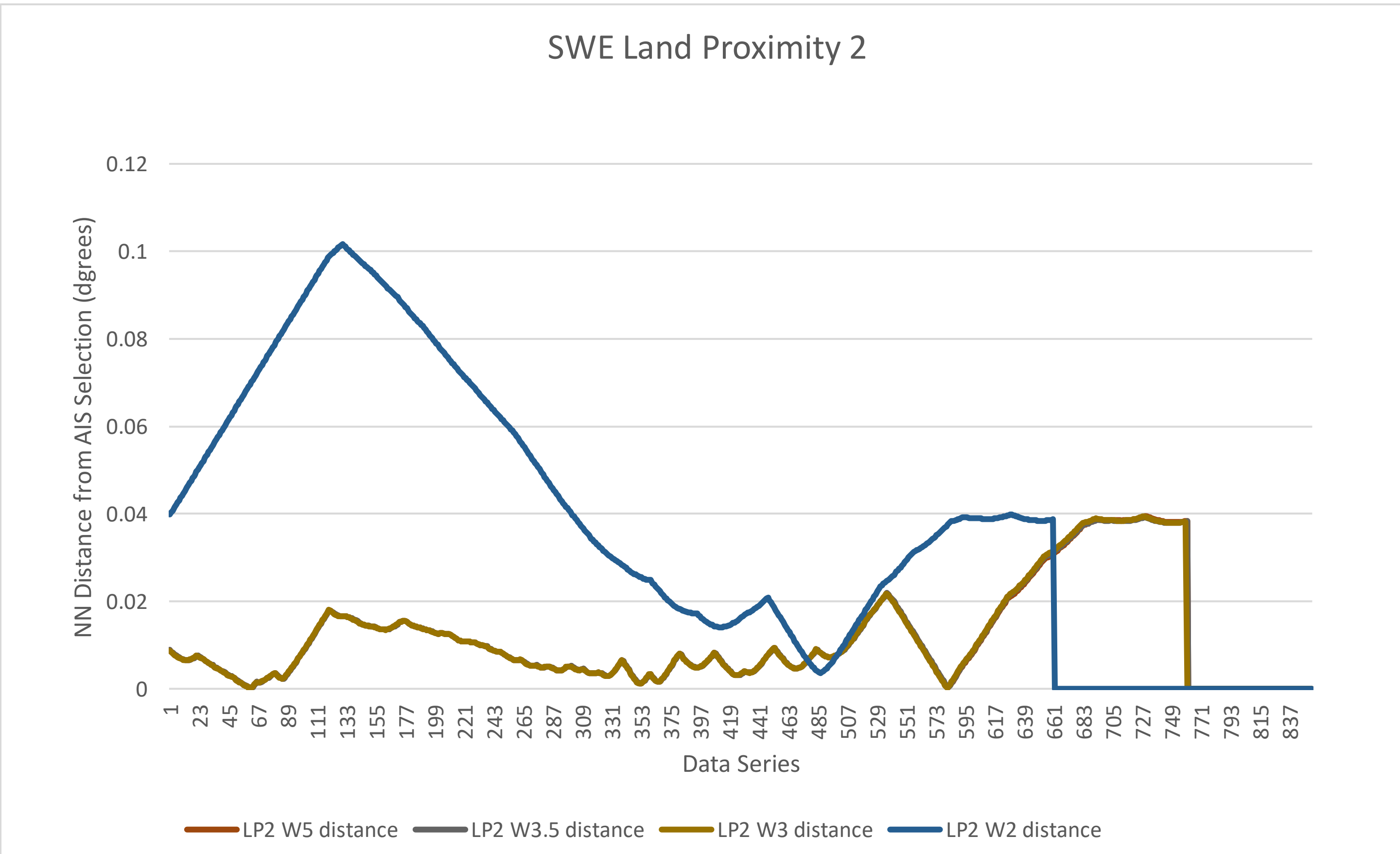


Figure 3.3.1 Ship Wise Errors Results Land Proximity 2

LP5 SWE results are less detailed than heatmap analysis but provide some insight. Although LP5W2 and LP5W1.5 have very similar routes, LP5W2 scores better for the middle portion of the simulation. It can also be identified the point at which LP2 and LP5 models differ most from LP3. The same areas of weak performance in the middle of the simulation for LP2 and LP5 are seen in heatmap analysis.

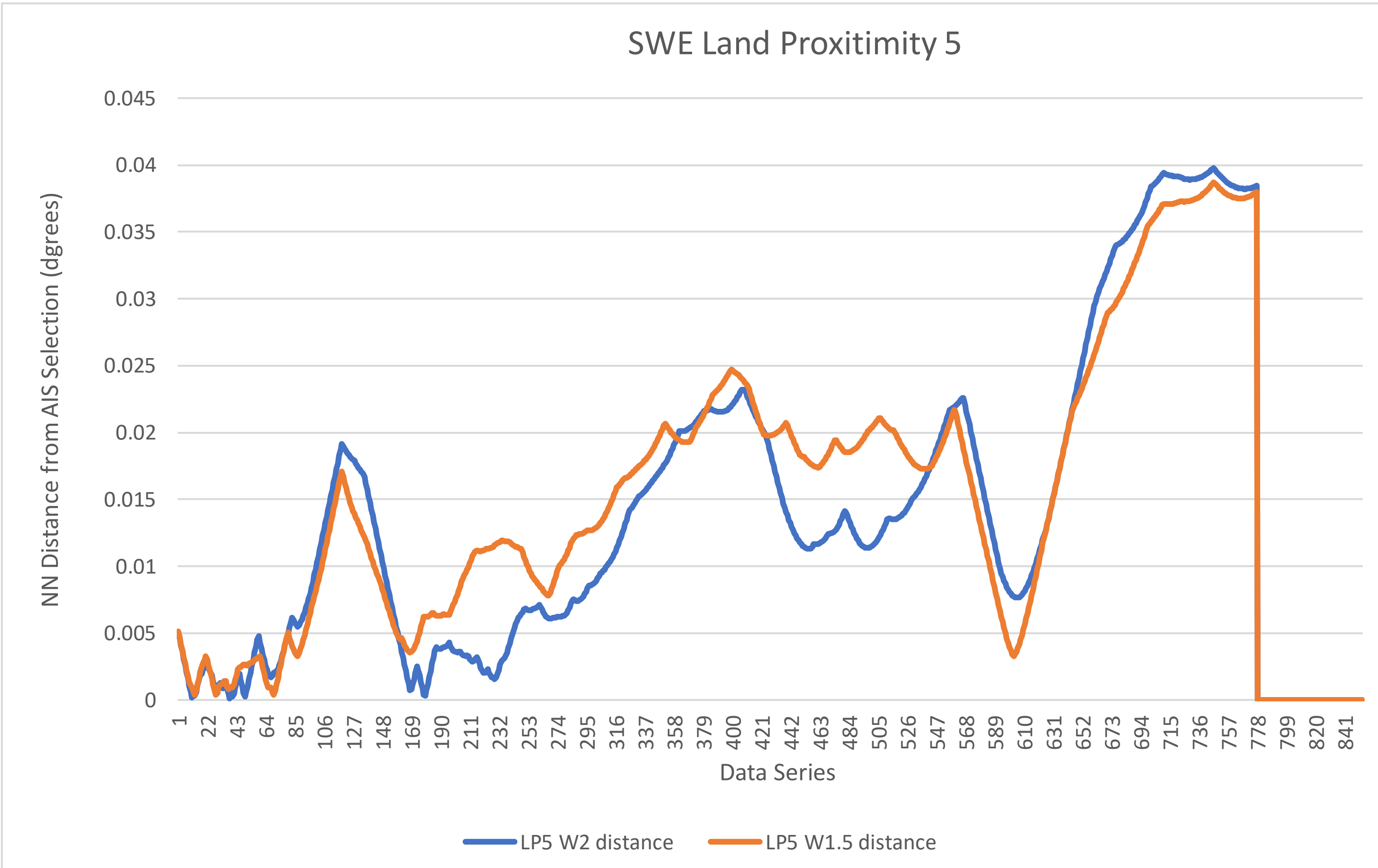


Figure 3.3.2 Ship Wise Errors Results Land Proximity 5

LP3 model iterations have the most variety in route selection when changing the LPW. However, SWE analysis does not provide the same level of detail. Major changes are all that can be identified when looking at LP3 SWE results. The dip in LP3W5 can be seen at approximately data series #260 but any more detail is much harder to determine. This corresponds with SPDHC analysis (Figure 3.2.1 Sequential Data 500M Heatmap Directional Land Proximity 3) where a spike in scores can be seen around data series #4900. This is again due to LP3W5 having a closer proximity to a small cluster of AIS data.

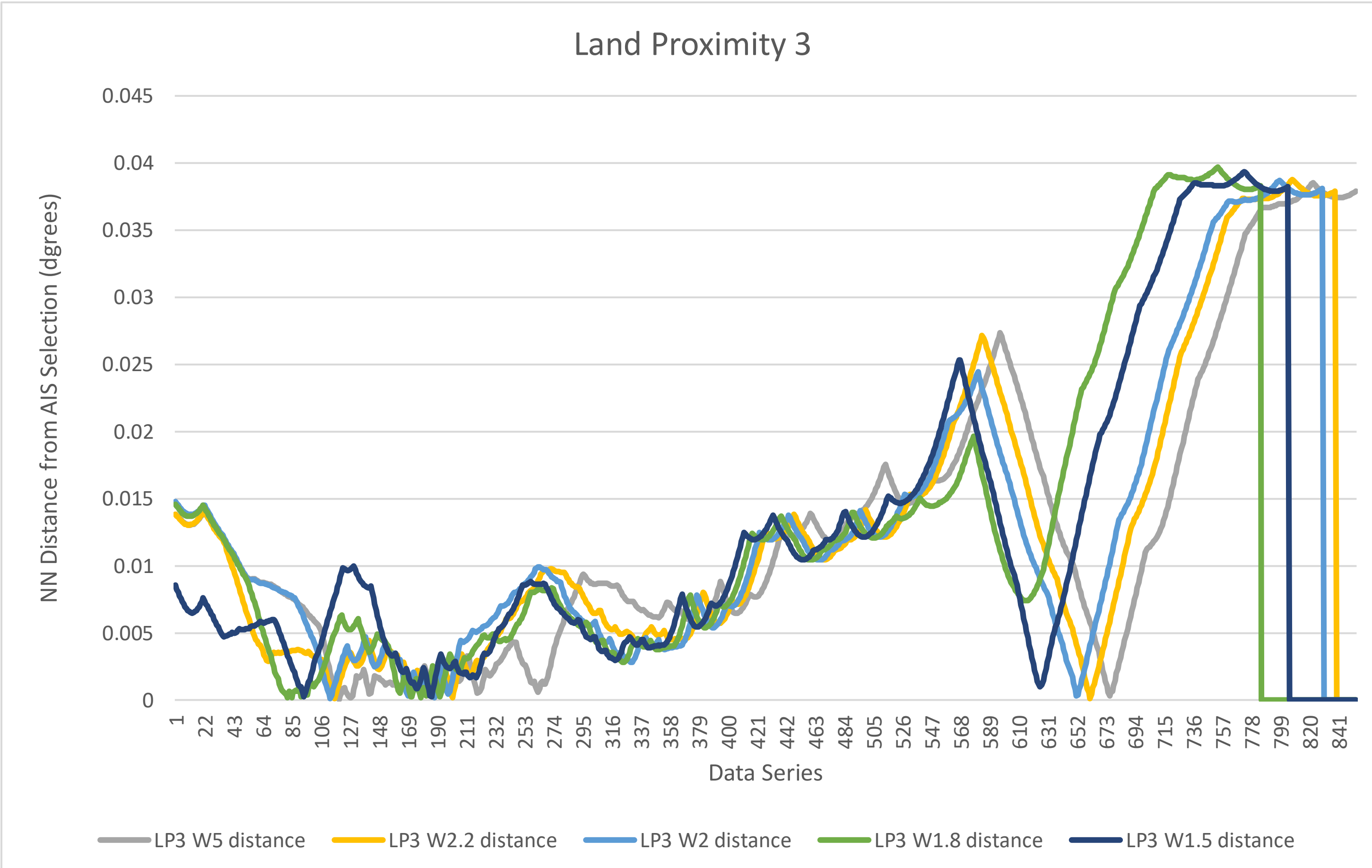
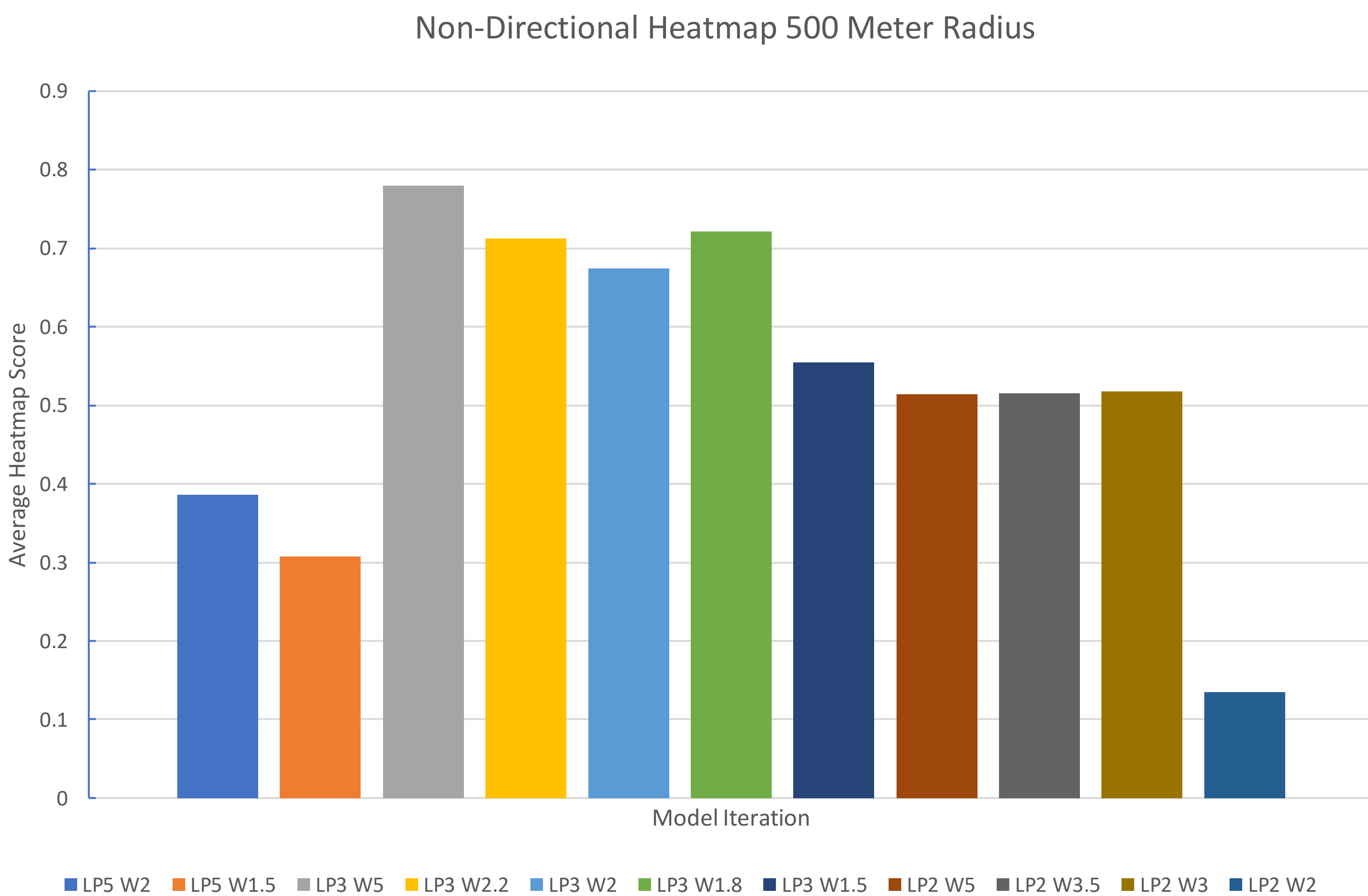
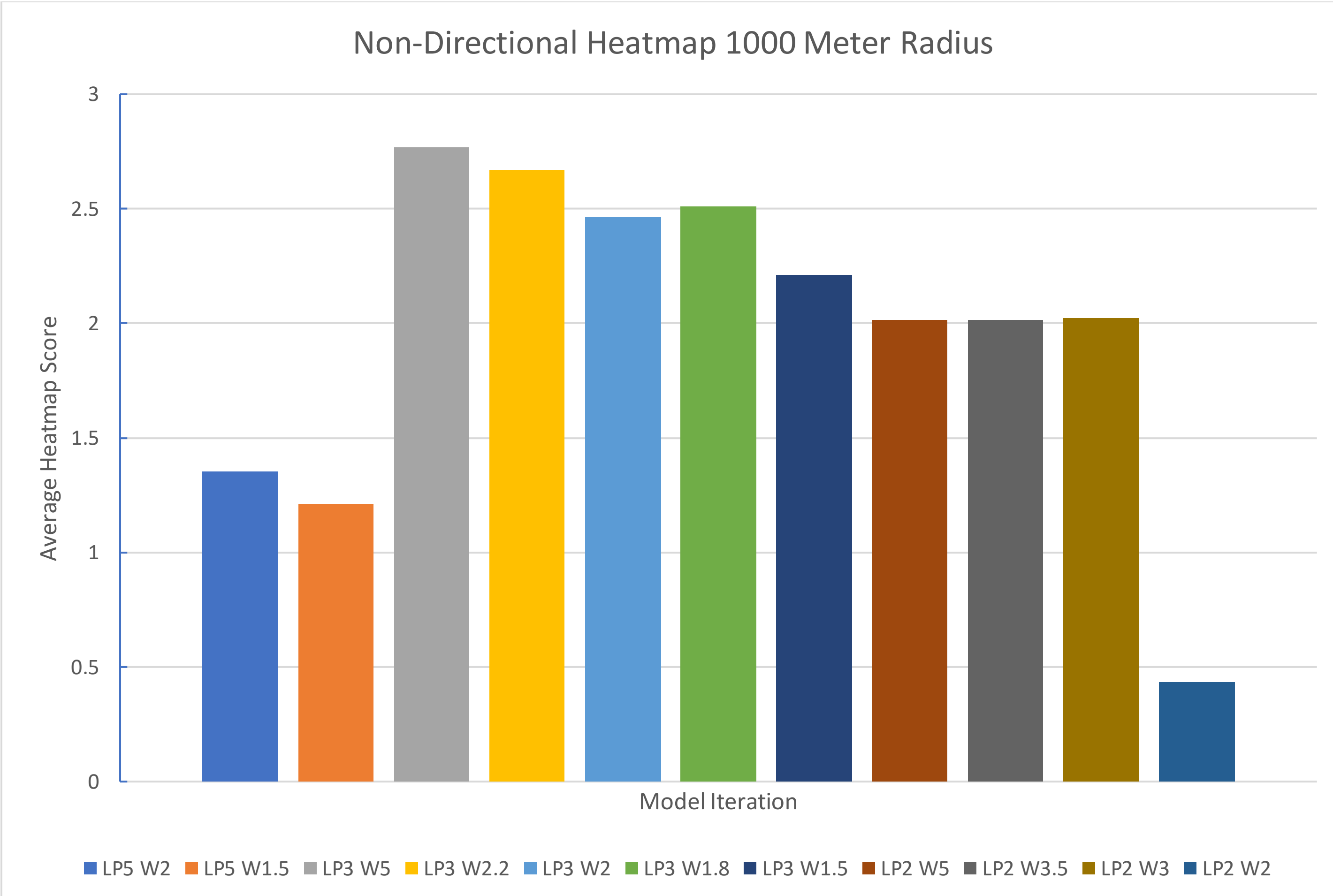


Figure 3.3.3 Ship Wise Errors Results Land Proximity 3

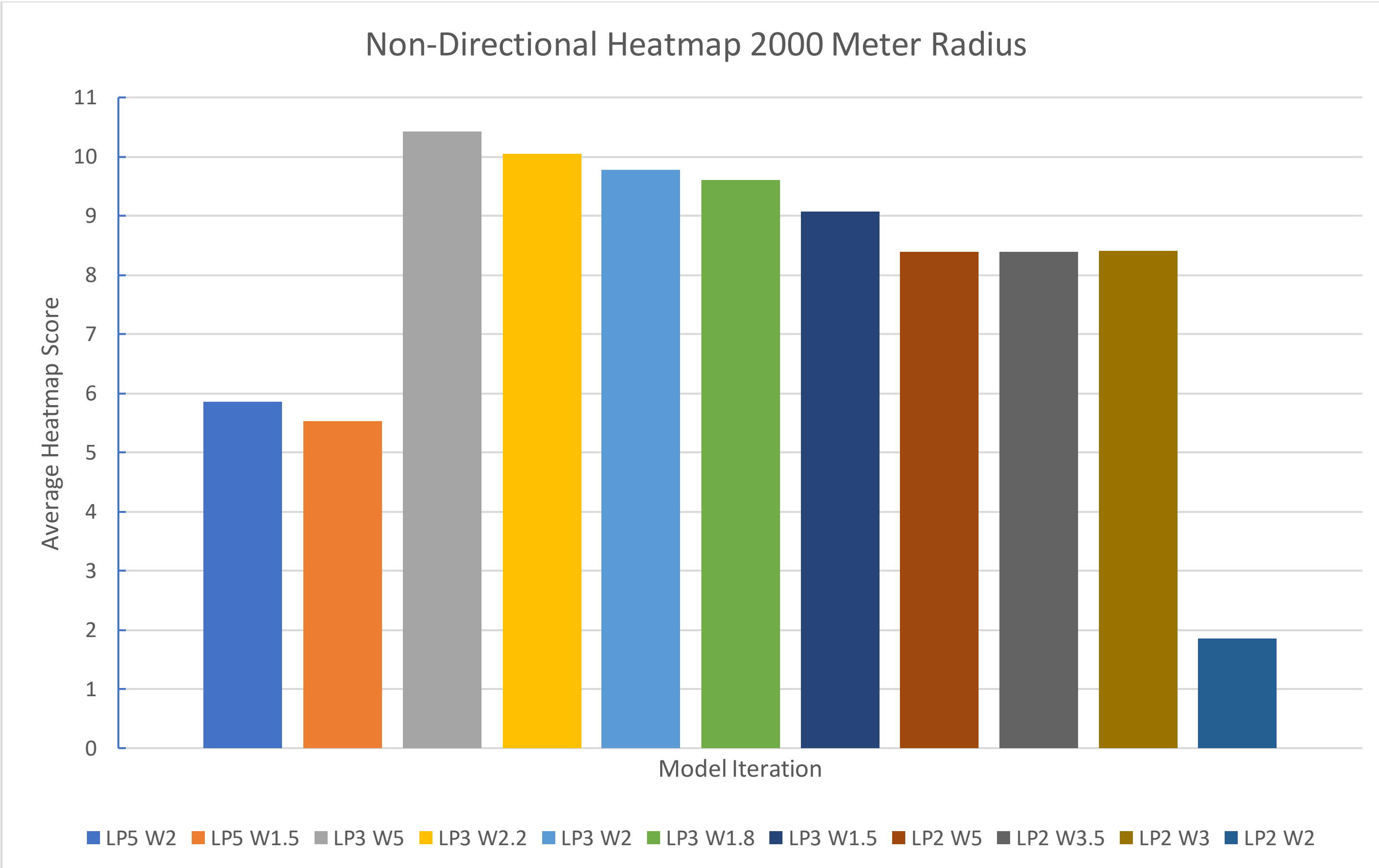
3.8 Multi-Directional Analysis



3..4.1 Non-Directional Heatmap Results 500 Meter Radius



3.4.2 Non-Directional Heatmap Results 1000 Meter Radius



3.4.3 Non-Directional Heatmap Results 2000 Meter Radius

Looking at directional vs non directional results provides insight into the importance of data cleaning to use big data effectively. The non-directional results score differently than directional results. The model performance order is significantly changed emphasizing the need for accurate AIS data. The directional analysis in the Port Metro case study is limited to inbound traffic as it provides more complex and pertinent information for port research. For example, the ability to have anchorage sites and scheduled port arrivals would not be necessary for outbound travel. During development, route selection tests were performed in the other direction (leaving Port Metro). With low LP and LPW values, the model would select a path (like LP2 LPW2) which is not viable for vessels and does not appear in the AIS data. Increasing the LP and LPW values was successful in altering path selection, simulating the correct route, south of Saturna island around Stuart Island and into the Haro Strait.

The LP and LPW values chosen for analysis are ideal for use in tight waterways. These values may not be suitable for study areas with less obstacles. Visual analysis of model iterations and heatmap results show that all the model iterations fail to accurately navigate the turn around Saturna Island. At the end of the simulation, all model iterations turn into oncoming traffic and do not follow directional traffic lanes. Increased scores at this location are the primary reason for the performance differences when comparing directional and non-directional results. Once ships travel past Saturna Island, the waterway opens without as many islands or obstacles to influence route selection. As a result, the model iterations take a more acute angle when navigating the island, traveling along incoming traffic lanes.

Future analysis could compare outgoing traffic to model results to see how the MSRM performs. This would provide insight into the entire journey of vessels travelling to port. Furthermore, inspecting inbound and outbound data can provide detailed analysis at the most complex locations during navigation (i.e., the turn at Stuart Island in the Port Metro case study). Without much space, route choice must be precise to stay in traffic lanes and avoid obstacles. The LP values would likely remain the same for both directions of traffic as the width of the channel is the same. However, LPW values require slight changes to ensure route selection with good directional performance. Specific functionality to identify oncoming traffic lanes as an obstacle is not implemented in the MSRM but refining LP and LPW parameters could provide adequate representation of simultaneous traffic lanes. Other solutions would include validating one direction of traffic and using the results as obstacles for simulating oncoming traffic. This would require changing the approach to selecting LP and LPW values for a given waterway but could ensure model results for both directions of traffic.

3.9. Anchorage location and scheduling

A vessel traffic service (VTS) is a maritime traffic monitoring system that is designed to improve efficiency and safety. VTS authorities are responsible for creating logistic schemes regarding navigation rules and port procedures that provide effective and safe scheduling with maritime traffic growth, inbound and outbound traffic experience significant waiting times, causing negative environmental and economic impacts (Li, Zhang, Yang, & Wang, 2020). Optimizing traffic procedures is a priority for VTS operators and simulating anchorage scenarios can assist in the development and verification of anchorage scheduling. To determine how the MSRM can be used to experiment with anchorage scenarios, an assessment of anchorage locations and wait times was performed. With the ability to add many ports (final destinations) and anchors(waypoints), experimentation with number, location and scheduling of anchorages and ports is possible. The current list of available anchors for cargo vessels for PMV (69 anchors) was converted from a .csv (comma-separated value) file to a .shp (shapefile) file to be used in the model. Test runs were performed to determine if all anchorage locations could be reached with LP and LPW weight parameters used in prior analysis. With an LP of 3 and LPW of 5 all waterways leading to the 69 anchors were successfully navigated and with the find-exit function, all anchorage locations were reachable. Figure 3.5 shows the MSRM with all the anchorage locations. Wait times provide implementation of scheduling as port wait times dictate how long ships remain at anchor with priority given to ship id order. Anchorages also have wait times but are only implemented when the port or the next anchor in the list is occupied. Simulations could test adding new anchorages or removing old ones, providing an assessment on changing wait times. Fuel estimations could indicate the amount of fuel used at anchor and how it pertains to overall shipping efficiency. Precise location of anchors could provide information on scheduling efficiency while prioritising environmental impacts and public opinion of anchor location. In the case of PMV, the whale initiative would benefit from identification of anchor wait times and whale sightings to prevent vessels from idling near whale habitats. Adding basemap data regarding whale location data or implementing agents in Netlogo simulating whale movement could provide further research into the problem.

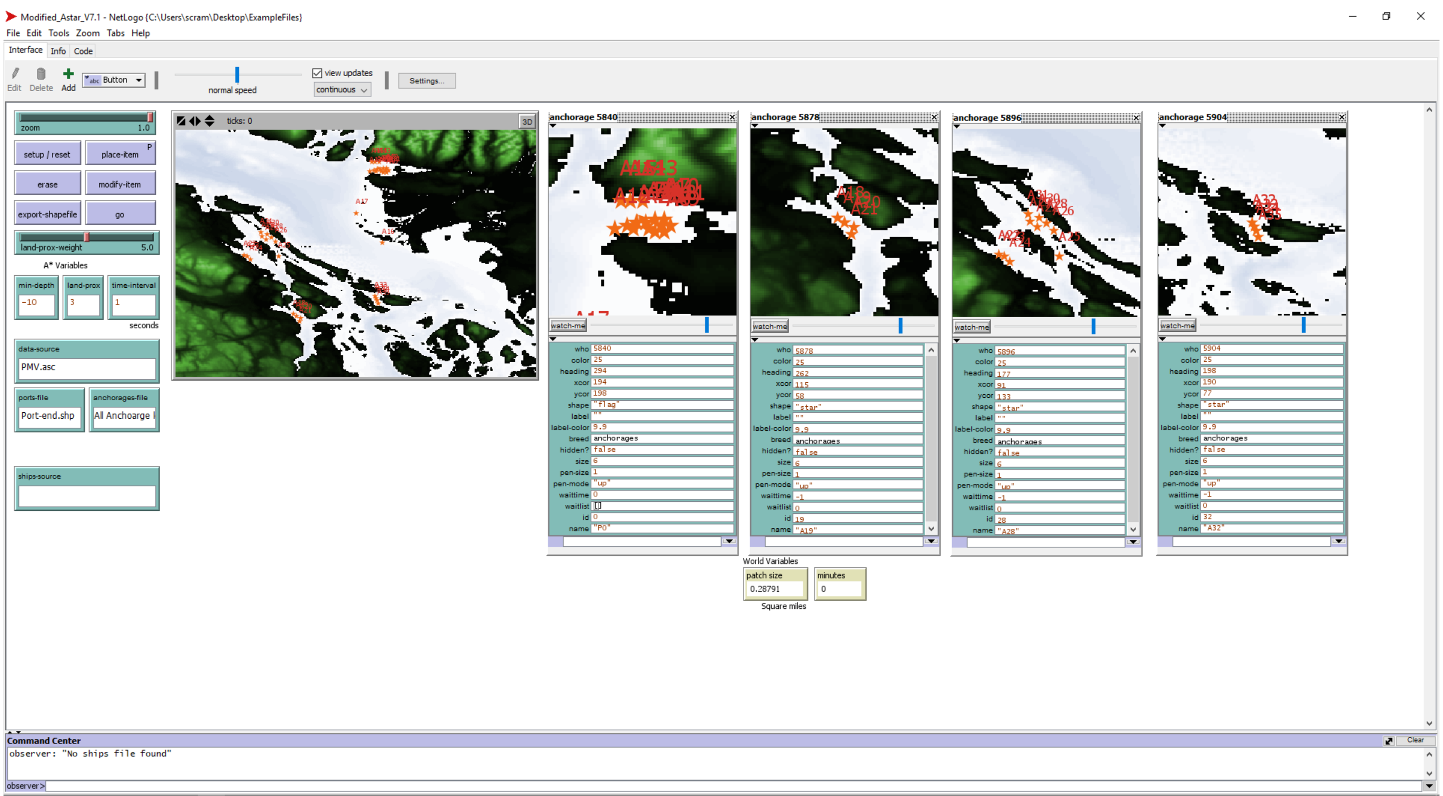


Figure 3.5 All Anchorage Locations

4. Discussion

4.1. Research Implications

In the case of maritime route simulation, the results have shown that the modified BFS algorithm used in the MSRM can replicate maritime route selection in confined waterways to a measurable degree of accuracy. Furthermore, it shows that large AIS datasets are necessary for accurately calibrating the model to optimize route replication. The research goal of replicating existing shipping pathways using a proprietary modified BFS algorithm is novel in the maritime shipping modelling field. Literature reviews such as "review of maritime traffic models from vessel behaviour modeling perspective" (Zhou, Daamen, Vellinga, & Hoogendoorn, 2019) categorize and summarize many publicly available models. Commercial models were excluded due to limited information provided to researchers. Their findings support that of this research, emphasizing that models without calibration via AIS data limit their applicability and can not accurately replicate historical ship movement(Zhou et al., 2019). Other researchers outside the maritime field support the finding that many ABMs aim to replicate real world phenomenon where using big data is an ideal way to calibreate and validate simulations (Kavak et al., 2018). The SWE assessment supports these findings by detailing the inadequacies of using small amounts of individual ship data for model calibration. Although information on ship routes and model performance can be identified using small amounts of AIS data, the results of this research have shown that detailed insight into route replication requires large AIS datasets.

Data quality is a primary focus when using large datasets to calibrate simulation results (H. J. Miller & Goodchild, 2014). Comparing directional vs non-directional results highlight the importance of using an appropriate dataset. The MSRM simulates inbound port traffic where the AIS dataset utilized includes both inbound and outbound traffic. The directional vs non-directional results show that filtering the dataset to isolate inbound AIS data provides better route anlaysis than using non-directional AIS data.

Addional findings from Zhou et al.(Zhou et al., 2019) provide insight into the modelling techniques used in the majority of maritime traffic models. In the 25 models they examined all but three represent the vessles as an agent. However, this does not mean agent based modelling was used as the simulation environment. Only a small number of models examined in the literature review utilize detailed manoueverabuluty with sub-modules – requiring the stregnths of ABM (Zhou et al., 2019). The only study assesed in the literature review that does utilize ABM is the piracy modelling by Vaenk et al. (O. H. Ondrej Vanek, Michal Jakob, Michal Pechoucek, 2011). Their simulation focused on the interaction of various vessel types in piracy scenarios (merchant vessel, navy vessel, and pirate vessel) and simplified sailing behaviour as it was not the goal of the research. The findings of the MSRM show that ABM can effectively be used to simulate maritime shipping with a single type of vessel. Other strenghts of ABM highlighted by this research include; the grid envirmonent featured in agent based modelling toolkits can easily traslate to GIS Raster imagery, ABM toolkits provide excellent primitives and extensions that allow for easier programming, ABM allows for autonomous behavior of vessles providing emergent phenomenon that may not be easily predicted.

The majority of current research in the maritime shipping modelling field examines detailed ship to ship interactions and protocols (including navigation procedures in ship encounters, fuel consumption based on rutter changes and engine usage, the effects of weather on maneuverability etc.) collision and accident modelling, piracy prevention and supply chain scheduling at port (Zhou et al., 2019). For example, Huang et al. (S. Huang, Hsu, Fang, & Song, 2016) use existing pathways to simulate complex traffic at large-scale hub ports. Simulating various types of vessels and effectively modelling the interactions amongst the vessels following existing protocol was the primary goal. Additional model application assesses the impacts of accidents causing partial lane closures under emergency scenarios. Many other studies focus on similar detailed ship interactions in different waterway scenarios (open-water and confined), with different vessel types and including the impacts of weather on vessel behaviour. Few studies model ship movement with the goal of replicating existing routing data. A commercial research project focusing on open water traffic density replication along predefined historical vessel trajectories is most like this research as its primary focus is replicating shipping routes. Known as MATRICS and developed by the Defence Research and Development Canada – Centre for Operational Research and Analysis(DRDC CORA) - The goal of the research is to “autonomously generate vessel tracks that tend to reproduce historical densities over time” (Hilliard & Pelot, 2012). Although similar in the fact that its replicating existing shipping routes, the focus is on open water way navigation through global shipping channels and not route choice behaviour in confined waterways. Furthermore, it does not use ABM and does not use a path finding algorithm for simulation instead travelling along predefined routes according to historical data via radar, traffic data and AIS data.

Commercial maritime shipping data is hard to acquire (Bourdon, Gauthier, & Greiss, 2007). When the data is publicly available it may be incomplete or inaccurate and often has many redacted fields removing important data. The implications of this research provide a starting point for conceptualizing how more advanced modelling can produce AIS data with the possibility of it being indistinguishable from real AIS data. This output can be used in the absence of real AIS data, or it can supplement current datasets, filling in gaps or providing additional information.

Port policy and management can utilize agent-based modeling to study the needs of the maritime domain (Li et al., 2020). The MSRM explores the use of anchorage scheduling to determine how the model can be used in a practical application. Vessel traffic services can experiment with a variety of anchorage related scenarios to inform policy. Furthermore, the modularity of the MSRM allows for experimentation with any waterway, providing application to other geographical contexts. Other case study specific examples include the Government of Canada's "Whale Initiative". Future experimentation can add new agents that represent whales. By analyzing whale sighting data, seasonal migration, and whale hotspots it is possible to examine the interaction of commercial shipping vessels and local wildlife.

4.2 Ship wise errors vs heatmap analysis

Heatmaps are a great way to examine the accuracy of model iterations to large AIS datasets. Big data allows for model results to be verified using a representative dataset with real ship location data. The ability to analyze and store large databases without the need for supercomputers has allowed more researchers to take advantage of big data. Random sampling was once the main strategy for dealing with information overload (H. J. Miller & Goodchild, 2014). Unfortunately, due to the nature of AIS data, individual sequential ship data from a single trip does not include sufficient point density to represent an accurate route. Due to AIS data collection protocol (SOLAS) and data quality, the frequency of location data collection produces less accurate representation of the path ships take in the study area.

Depending on the actions of the ship (speed, course etc.), data will be collected more or less frequently creating inconsistent gaps in the data. Having a larger data set allows for higher frequency of location data in the study area allowing for more accurate representation of AIS signals. This provides for a better assessment of model results. Furthermore, by using large AIS datasets, inconsistencies in ship routes due to; slight navigation changes, AIS location data accuracy and error, position of data collection, are reduced. This is visible when comparing SWE results to heatmap results.

The dataset includes 1 month of AIS data for cargo ships. Having access to more data was possible, due to computational limitations and processing efficiency the data set had to be limited to the chosen selection. More data would improve the heatmap density and allow for potential seasonal adjustments taken by ships to be analyzed. By analyzing the results, it would appear that the amount of data used was large enough for accurate representation.

Nearest Neighbour analysis requires manual selection of AIS data which may not be an accurate representation of ship routing. Human selection of data introduces inconsistencies and is less likely to provide the same level of detail. The most simplistic changes in model iterations i.e., LP2 changes and minor differences in LP5W1.5 and LP5W2, can be deduced from the SWE results. More detailed changes like those present in LP3 model iterations are not visible in SWE results. The changing heatmap radius and analyzing directional vs non directional results provide further insight into model iterations that is not provided by SWE analysis. Using several thousand AIS data points and heatmaps emphasize clusters of signals. These data clusters often occur when ships are turning. Model results in these scenarios are harder to produce so increasing the value of scores for these sections is useful. Data clusters also indicate areas more frequently travelled; SWE data selection does not provide the same level of analysis.

4.3. Limitations

The MSRM is a useful tools for simulating maritime traffic as described thoroughly in this research. However, limitations do exist regarding ABM, the MSRM, big data analytics and heatmaps.

4.3.1. ABM and MSRM Limitations

Identifying ideal LP and LPW weight values for a given waterway requires human input with trial and error. This limits the effectiveness of the MSRM as selecting ideal parameters can be difficult. Human input can be inherently flawed and less reliable for determining model parameters compared to automated computer functions (A. Miller, 2019). Selecting model parameters for appropriate simulations is tedious as many possibilities must be tested. Only then can ideal LP and LPW values be selected with confidence. Valuable model iterations in certain waterway scenarios may not be tested during initial model selection. This can limit analysis as crucial model functionality may be gathered from analysing model iterations that are not tested.

The MSRM requires large AIS datasets to support detailed analysis of vessel route replication. This is exemplified when comparing SWE analysis with a small selection of AIS data, to heatmap analysis with large datasets. This limits the opportunity to simulate other areas. Rough verification of model parameters can be performed using basic shipping routes. This information is readily accessible on the internet for all major ports. This allows for basic comparison of model iterations without the need to access restricted, expensive, AIS datasets.

However, verification and calibration of the MSRM is limited without big data analytics. Results in the Port Metro case study support findings that the MSRM is capable of accurate route replication. Without AIS data from other ports the MSRM lacks reliability as only one case study can be performed.

The complex nature of ABM and potential for diverse waterways in the MSRM can provide inconsistent results. With many calculations and unknown details to primitives, identifying where changes in programming affect changes in model output can be complicated. Netlogo supports effective troubleshooting by identifying agents and areas of code when an error occurs. This can still be confusing as errors can occur with similar model parameters executed successfully in prior models.

The modified BFS algorithm for maritime traffic provides efficient and accurate route replication. The find exit error mentioned previously is a limitation of the MSRM. This error prevents the model from correctly functioning in all waterway scenarios with certain LP and LPW parameters. Model iterations can incur errors where they are unable to proceed. The model will not identify this as an error, causing the model to run indefinitely until halted by the operator. This error only occurs when LP and LPW values are higher than needed for the given waterway. This will rarely occur as LP and LPW values can be modified with the theoretical exception of specific waterway scenarios with large changes in channel width.

The scale of the map in Netlogo determines the level of accuracy when compared to AIS data. With computational demand proportional to the number of patches in the waterway, and limitations to world size in Netlogo, the scale of the basemap is limited. This can provide inaccurate results as model production may follow patches that do not align with AIS data at a higher resolution. This will also change BFS calculations as the location of boundaries are crucial for determining the shortest path. As a result, the MSRM produces data that can appear to travel on land when overlaid by continental shape files or AIS data.

4.3.2. Big Data and Heatmap Limitations

Big data analytics is crucial for verifying and calibrating the MSRM. There is a limit to the amount of data needed to provide accurate analysis of model results. With too much data, computational times suffer, data cleanup can be more complicated, and results may not benefit from the increase in data. A limit exists to how much data is needed for the application of the MSRM. This also applies to other computational models as too much data can cloud analysis (Sui, Goodchild, & Elwood, 2012).

GIS tools are useful for cleaning large AIS datasets. The process involves selecting vessels based on various data values and manually removing them from the dataset. In the Port Metro example, vessels with speeds of zero are removed to prevent ships that are idle or at anchor from affecting heatmap calculations. Directional analysis removed vessels with headings indicating outbound port travel. However, some data can be removed that is pertinent for the research question. For instance, accidents or abnormal ship behaviour can provide useful information related to the research objective. By identifying this data as not useful, anomalies crucial for detailed waterway analysis may be neglected. This also applies to noise or spurious data that is removed manually due to its proximity to the expected path.

Heatmap radii greatly affects model results. There is not one suitable setting for all scenarios, and it is hard to determine what heatmap radius provides the best representation of model comparison. Smaller heatmap radii emphasize slight deviations from AIS data. Larger heatmap radii are more generous, providing scores to model iteration points that fall further away from AIS data. Where and when heatmap radii should be used depends on detailed analysis of the waterway. The Port Metro Case study shows that the radii can change the rankings of model iterations. LP values tend to remain the same as altering values creates more drastic simulation changes, and consistent results regardless of radii is to be expected. LPW rankings are more complicated and small changes to LPW create slight differences in simulation results. The heatmap radius will change LPW rankings and determining how trends in LPW change with heatmap radius is complicated.

4.4. Model Improvements

Current model results provide the necessary insight to examine the use of big data analytics and ABM for Maritime simulation scenarios. Potential model improvements include reducing human input, improving model efficiency, and reducing errors.

4.4.1. Reduce Human input

The need for human input to select suitable LP and LPW values for analysis can be tedious. Applying the MSRM to a waterway requires a preliminary assessment where manual selection of parameters is needed. The process can become more difficult with complex waterways. Incorporating the ability to automate initial parameter selection would reduce the time required to select LP and LPW values and avoid trial and error when running simulations. This would limit the need for human input and optimize the initial waterway assessment. Developing a script to analyse AIS data along the simulation route, information could be gathered regarding the narrowest channel or the area with the smallest number of pixels between opposite or adjacent land borders. This could provide a starting point for selecting LP values that would be ideal for further analysis.

The process of comparing, importing/exporting data and running simulations could be incorporated into one operation, automating the process. Although hypothetical, this would require extensive programming that would need to run model iterations automatically based on results from computing heatmap scores. By calculating the shortest path algorithm without any modification, (default BFS) the results can be compared to large AIS datasets using heatmap analysis. By repeating this process with increasing LP and LPW values, an ideal heatmap score can be reached indicating optimal replication of the AIS data. The average heatmap score and SPDHC can be determined for different model iterations and inform parameters for new model output. Netlogo's online "Hub-net" could be used to update LP and LPW values. This would allow for python scripts or other tools in GIS software to communicate with Netlogo to provide a way to automate model creation.

4.4.2. Improve Model Function

The modified BFS algorithm is not suited for heterogenous waterways. Transitioning from open water navigation with few islands to tight waterways with many obstacles would require vastly different LP values to replicate AIS data. The Port Metro case study provides an example of this, as simulations fail to accurately replicate AIS data at the end of the simulation. Addressing this problem with current model functionality would be complicated and require significant changes to core model functions. A simpler solution would be to run several simulations at different portions of the route. With a greater LP, more patches on route to the destination would be affected, producing path likeness more accurately for open water navigation. For tight waterways, reducing LP and increasing LPW is needed. For instance, LP10 LPW2 for open waterways and LP2 LPW3 for tight sections. Creating berth and anchorage sites with GIS software, the user can ensure that the MSRM simulates all sections of the waterway. Although this would add significant time to process and analyse, combining the various models would allow for route replication in changing scenarios and ultimately, more accurate path replication.

The Find-Exit function is crucial for allowing vessels to travel to port regardless of its proximity to land; if the destination is still "reachable" (above the min depth specified). This allows for LP and LPW values needed for navigation, without restricting access to port or anchorages near land. Without this function, errors would occur where the ship would be unable to reach the destination. Hypothetically a situation could exist where the model would encounter a similar situation without being close to the destination. This would create errors and not allow the ship to continue even if the specific channel is used in real-life navigation. Implementing a function that occurs in the movement stage of the model, similar to "find-exit", could identify areas where the simulation would get stuck. However, without any knowledge of real data it would be impossible to know where this is necessary or where altering LP and LPW values would provide the same adjustment needed to reach the destination. A possible solution would be to recalculate BFS values if the ship returns to the same pixel more than once. Using the point at which the model is stuck to recalculate BFS values from the destination, a find-exit function could be applied to the start point with the destination being a next most favourable path that does not lead to the same point.

The precise movement of simulation results appear to have many small route adjustments that are not present in AIS data. Due to the frequency of point capturing in Netlogo, slight changes in route choice reduce the smoothness of the path. Increasing the tick duration of the model would reduce density of points when collecting footprints and produce smoother lines when resampling in GIS software. However, this would reduce the amount of route selection detail when observing model results.

4.5. Analysis changes

4.5.1. Simulate Multiple Locations

During model development, other waterways (Victoria Harbour in HK and the Gulf of California) were tested to ensure correct model performance in all scenarios. Both examples were more simplistic as less obstacles and a more direct choice of path selection allowed for less modification of BFS values to navigate. Without AIS data from these regions a full analysis was not possible. Acquiring AIS data from other ports would allow similar case studies to be performed. This would improve the reliability of the MSRM, as a variety of tests can increase confidence in strong model performance. Other ways to improve model analysis would include, analyzing a greater variety of heatmap radii, applying different heat map estimation calculations, increasing the amount of model iterations used in analysis, redoing SWE calculations with a different selection of AIS data. Most of these solutions improve model analysis by increasing confidence of the findings with more data and more research angles to explore.

4.5.2. Simplify SPDHC Comparison

SPDHC provides information about model performance at various points of the simulation; Without this, detailed analysis of model performance would not be possible. Comparing SPDHC data points in graphs or spreadsheets to locations on the map or model output can be challenging. Manual reference of data series ID and point ID is required to identify where the data aligns with simulation points. Creating a function in R or other programming languages designed for statistical computing and graphics, could help automate the process. This would simplify model iteration analysis by allowing the user to compare areas of interest quickly and confidently in the data series or map output.

4.5.3. Linear Trajectory Analysis

AIS point data can be seen as a representation of trajectories each ship takes in a continuous journey. This allows static position or trajectory density maps to be produced. Heatmap analysis uses point density to determine model results however, this can be problematic as heatmaps may be difficult to interpret especially when quantifying the intensity of maritime traffic (Tixerant, Guyader, Gourmelon, & Queffelec, 2018). To deal with this, an estimation on ship trajectories as line data can be an effective way to compare MSRM model results to AIS data. This would allow model routes to be represented as trajectories addressing some of the key issues with point data. For example, static ships sending a greater number of messages, disproportionately affecting the distribution of point data density. However due to the quality of much of the individual vessel data and the small scale of the case study region (PMV), larger gaps in AIS data can negatively impact many vessel trajectories. Many studies primary focus is determining the trajectories of ships between AIS data points. For example Borokowki (2017) presents a algorithm to predict ship movement trajectories (Borkowski, 2017).

4.6. Additional functions

The simplicity of the BFS algorithm and efficient implementation in the MSRM allows for quick computation with thousands of nodes (patches in Netlogo). The MSRM can add as many vessels as there are patches (in theory), without the need to calculate each agent's shortest path to various destinations. This is very important for further implementation as it allows for efficient BFS calculations for many destinations without increasing computational demand based on the number of vessels. This also allows for separation of route calculation and movement functionality. Without reducing the execution speed of Netlogo, simulations complete the entire ship journey (>100,000 ticks) faster than the ability to update the image; requiring a reduction in execution speed to observe simulations live. The headroom in the move section is intentional, providing the option to add additional functionality with ease. The operations in the MSRM were designed with the intention of having additional simulation criteria. Significant effort into ensuring compatibility with increased movement complexity and port/anchorage interaction was a priority during development.

4.6.1. Altering Ship Movement

Addressing the potential "find-exit" error during navigation would provide similar functionality needed to incorporate ship to ship interaction. Large BFS values can create a scenario where vessels cannot proceed. The simplest solution to this problem would be to alter a vessel's course to return to the desired trajectory from its new location (Wu, Peng, Ohtsu, Kitagawa, & Itoh, 2012). The same requirement is needed to provide ship to ship interaction and other shortest path deviations due to weather or other obstructions.

Netlogo provides agents the ability to identify various parameters including the location of other agents. Avoiding moving obstacles can be complicated as the agents must be aware of the ship's future location and direction of travel. Using the list of patches assigned to each agent provides information regarding future patch selection. However, determining where and when vessels meet may be difficult.

Assigning priority to vessels based on real traffic procedures provides a set of rules to be followed in various traffic scenarios. This includes providing weather data and other events that may change vessel course. Following applicable traffic rules and weather events, ships could alter their route and continue shortest path calculations from their new location. The new location would need to adhere to the “reachable” clause ensuring it remains safe for travel. This requires reducing speed parameters to represent stopping and slowing down to provide safe passage. To replicate realistic movement, data on rudder motion, wind speed and direction of wave/currents is necessary (Wu et al., 2012). Furthermore, multiple ships interacting significantly increases complexity. These problems are difficult to describe and require extensive research and programming to provide even the most basic level of ship interaction and on-the-fly route selection.

5. Conclusion

Analysis of the MSRM proves that a modified BFS algorithm designed for maritime navigation, can accurately and efficiently produce route selection that replicates cargo vessel AIS data. Identifying land or shallow water using bathymetry data and GIS analysis, provides boundaries for manipulating BFS values and ultimately route selection. In the MSRM, LP is used to determine the number of nodes in proximity to land that will be modified. LPW acts as a multiplier, increasing BFS values linearly as the distance from land increases. These simple modifications allow for efficient functionality and quick computation. Using various analysis techniques, details how different parameters for route selection affect replication of real shipping routes. Analysis using QGIS and Excel prove the MSRMs ability to accurately replicate shipping routes in hard to navigate waterways and provides insight into the detailed functionality of the algorithm.

AIS data is necessary for comparing simulations to real data. Detailed examination of model performance can be determined by using big data analytics in the form of heatmap analysis, not possible without large datasets. SWE analysis using a small amount of hand selected AIS data, highlights the benefits of big data when analyzing maritime route replication. Heatmaps utilize big data effectively by showing where vessels travel more frequently providing enhanced detail over basic shipping routes. Heatmaps also reduce errors present in big data as outliers do not significantly affect simulation comparison. Directional versus non directional analysis emphasizes the need for extensive data cleaning when using big data. Heatmap analysis and SPDHC present detailed insight into changes model parameters have on route selection.

ABM tends to replicate phenomena where real world data can be used to validate and calibrate simulation models. Big data is particularly useful as it provides many options for analysis and increases confidence in the results. However, Big datasets increase the difficulty of model comparison as extensive data cleaning can be required. Furthermore, limits to the amount of data often exists when using various software such as ABM toolkits. The results of the MSRM show that analysis using big datasets increases the detail of model performance characteristics. Determining exactly how much data is needed to best replicate vessel behaviour is difficult. The amount of detail in comparing AIS data to model results will reach a limit before any increase of data may not have a significant change if any on route analysis. Future analysis could experiment with incrementally reducing the amount of data used for comparison. By reducing the amount of data, it is possible to determine how much is required to provide similar results to that of this study.

In the Port Metro case study, narrow channels and many islands provide complicated route selection, ideal for testing the MSRM. The LP and LPW parameters will differ amongst waterways but in the case of Port Metro, a LP of 3 was most effective. LPW rankings were less predictable and depended heavily on the heat map radii used for analysis. LP values higher than 5 failed to create accurate paths with such narrow waterways in the simulation region. LP values lower than 3 provided degrading performance with smallest LPW values failing to choose the correct route (LP2 LPW2). Overall, results show that larger LP and LPW parameters increase the amount of obstacle avoidance. This was hypothesized during algorithm development and is supported by the findings in the Port Metro case study. When ports or anchorages exist within LP boundaries, additional modification is necessary for vessels to reach their destination. The “find-exit” function allows vessels to reach the destination regardless of proximity to land and is crucial for a model designed to simulate maritime port navigation. However, simulations have the possibility to run into a similar error when high BFS values cause the vessel to get stuck. Solutions to this problem can provide potential applications for further research. Altering routes during the movement phase of the simulation would require extensive programming. Possible solutions would present functionality necessary for mid-navigation obstacle avoidance. This would provide the ability to include additional functions to the MSRM, for example, ship to ship interaction, weather, and other safety assessments.

Route calculations and movement procedures are programmed as separate functions for a variety of reasons. First, it allows for simple addition of waypoints (anchorages and ports) and ships by importing GIS data or with manual selection on the Netlogo interface. Researchers can test various scenarios by adding agents as they wish, without requiring additional models. It also enables changing parameters for different waypoints, as BFS calculations are accomplished after each waypoint is created. Isolating movement parameters allows for fast execution of route selection. This provides potential experimentation of maritime scheduling, as fast execution time allows for quicker, more efficient simulations. Reducing computational demand in the movement section leaves ample space to add more complex movement experimentation.

Netlogo and other ABM toolkits provide a useful and simplistic approach to modelling that assists in the creation of the MSRM or similar models. Without ABM toolkits, programming the MSRM would be out of scope for this research and require significantly more expertise and time. ABM toolkits simplify the process but also create difficulties for researchers.

Primitive function is often not accessible and limitations in Netlogo or other ABM toolkits can be harder to overcome or avoid without full control of all functions.

GIS analysis is a useful tool for presenting and analysing model and AIS data. It allows for connection to Netlogo using extensions and greatly improves the ability to analyse model results. QGIS accommodates visual representation of model results and the creation of maps for output. It also provides heatmap calculations that are exported to Excel for further graphing and analysis.

8. MSRM Code

```
extensions [
  gis
]

globals [
  lastID
]

breed [borders border]
breed [waypoints waypoint]
breed [ships ship]
breed [anchorages anchorage]
breed [banners banner]
breed [footprints footprint]

patches-own [
  elev
  land
  cost
  reachable
]

waypoints-own [
  path
  speed
  A-fuel-cons
  B-fuel-cons
  name
]

ships-own [
  path
  speed
  A-fuel-cons
  B-fuel-cons
  lot-by-timeInterval
  totaltime
  last-patches
  current-layer
  fuel-cons
  name
  landed
]

footprints-own [
  shipID
  time
  shipspeed
  fuelcons
]

anchorages-own [
  waittime
  waitlist
  id
  name
]

;;;;;;;;;;;;;; setup ;;;;;;;;;;;;;
to setup
  ca
  reset-ticks
  draw-map
  import-elements
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; create (anchorages, obstacles & ships) ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to place-item
  let listOpc ["ship" "port" "anchorage" "obstacle"]
  ifelse any? anchorages with [waittime != -1]  [ set listOpc but-last listOpc ] [ set listOpc but-first listOpc ]
  let opt user-one-of "Select kind of item" listOpc
  while [mouse-down? = false] []
  if mouse-down?
  [
    ask patch mouse-xcor mouse-ycor
    [
      if land = true or elev > min-depth  [ user-message "No valid position" stop ]
      if opt = "ship"       [ place-ship ]
      if opt = "port"       [ place-port ]
      if opt = "anchorage"  [ place-anchorage ]
      if opt = "obstacle"   [ place-obstacle ]
    ]
  ]
end

to modify-item
  let listOpc ["ship" "port"]
  let opt user-one-of "Select kind of item" listOpc
  if opt = "ship"       [ modify-ship ]
  if opt = "port" [ modify-port ]
end

to modify-ship
  let listOpc []
  ask waypoints [ set listOpc lput [name] of self listOpc ]
  let option user-one-of "Select item" listOpc
  let x one-of waypoints with [name = option]
  ask x
  [
      set path []
      let opt "A"
      while [ first opt = "A" ]
      [
        set opt user-one-of "Select next anchorage or port" [label] of banners
        set path lput read-from-string but-first opt path
      ]
      set speed read-from-string user-input "Average speed (knots)"
      set A-fuel-cons read-from-string user-input "'A' parameter. Fuel_Cons = A*x^2 - B*x"
      set B-fuel-cons read-from-string user-input "'B' parameter. Fuel_Cons = A*x^2 - B*x"
  ]
end

to modify-port
  let listOpc []
  ask anchorages with [waittime != -1] [ set listOpc lput [name] of self listOpc ]
  let opt user-one-of "Select item" listOpc
  let x one-of anchorages with [name = opt]
  ask x
  [
    set waittime read-from-string user-input "Wait time (in minutes)?"
    set waittime (waittime * 60)
  ]
end
```

```
to place-banner
  hatch-banners 1
  [
    set size 0
    set label-color red
    create-link-from myself
    [
      tie
      hide-link
    ]
    move-to one-of in-link-neighbors
    set heading 50
    fd 15
    if-else [waittime != -1] of myself
    [ set label (word "P" [id] of myself) ]
    [ set label (word "A" [id] of myself) ]
  ]
end

to place-anchorage
  sprout-anchorages 1
  [
    set shape "star"
    set size 6
    set color orange
    set id (count anchorages - 1)
    set name (word "A" id)
    set waittime -1
    place-banner
  ]
  find-shortest-path-to-ships
end

to place-port
  sprout-anchorages 1
  [
    set shape "flag"
    set size 6
    set color orange
    set id (count anchorages - 1)
    set name (word "P" id)
    set waittime read-from-string user-input "Wait time? (in minutes)"
    set waittime (waittime * 60)
    set waitlist []
    place-banner
  ]
  find-shortest-path-to-ships
end

to place-obstacle
  while [ mouse-down? ]
  [
    ask patch mouse-xcor mouse-ycor
    [
      set elev 100
      set pcolor black
      set land true
      ask neighbors
      [
        set elev 100
        set pcolor black
        set land true
      ]
    ]
  ]
end


to place-ship
    if count anchorages = 0 [ user-message "No port/anchorage setted" stop  ]
    sprout-waypoints 1
    [
      set size 4
      set shape "circle"
      set color orange
      set name (word "S" (count waypoints - 1))
      set path []
      let opt "A"
      while [ first opt = "A" ]
      [
        set opt user-one-of "Select next anchorage or port" [label] of banners
        set path lput read-from-string but-first opt path
      ]
      set speed read-from-string user-input "Average speed (knots)"
      set A-fuel-cons read-from-string user-input "'A' parameter. Fuel_Cons = A*x^2 - B*x"
      set B-fuel-cons read-from-string user-input "'B' parameter. Fuel_Cons = A*x^2 - B*x"
    ]
end

to place-ships
  erase
  ask waypoints
  [
    hatch-ships 1
    [
      set speed [speed] of myself
      set A-fuel-cons [A-fuel-cons] of myself
      set B-fuel-cons [B-fuel-cons] of myself
      set name [name] of myself
      set landed false
      set fuel-cons 0
      let mph speed * 1.150779 ; mph
      set lot-by-timeInterval (mph * time-interval) / 3600
      output-show (word "Speed: " speed " knots. " lot-by-timeInterval " miles every " time-interval " tick")
      set totaltime 0
      set path [path] of myself
      set current-layer 0
      set size 3
      set shape "boat top"
      set color red
      set pen-size 1
    ]
  ]
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; A* ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to find-shortest-path-to-ships
  set lastID count anchorages - 1
  ask patches with [land = false]
  [
    set cost lput 0 cost
    set reachable lput false reachable
  ]
  set pcolor gray
  set reachable replace-item lastID reachable true
  calculate-costs self
  if pcolor = red [ find-exit 0 ]
  ask patches with [land = false][set pcolor scale-color blue elev -3500 0]
end

to find-exit [new-cost]
to find-exit [new-cost]
  let nextPatch 0
  ask self
  [
    set pcolor green
    set cost replace-item lastID cost new-cost
    if any? neighbors with [pcolor = gray]
    [
      stop
    ]
    set nextPatch min-one-of neighbors with [last reachable = true and pcolor != green] [last cost]
    ask nextPatch
    [
      find-exit new-cost + 0.1
    ]
  ]
end

to calculate-costs [source-patch]
  let current-patch 0
  let open []
  set open lput source-patch open
  let current-cost 0
  while [ length open != 0 ]
  [
    set current-patch item 0 open
    set open remove-item 0 open
    ask current-patch
    [
        set current-cost (last [cost] of current-patch) + 1
        ask neighbors4 with [ land = false  and elev <= min-depth and ( (pcolor != red and last cost > current-cost) or last reachable = false) ]
        [
          set pcolor gray
          set reachable replace-item lastID reachable true
          set open lput self open
          set cost replace-item lastID cost current-cost
        ]
        set current-cost last [cost] of current-patch + 1.4142
        ask neighbors with [ land = false and elev <= min-depth and ( (pcolor != red and last cost > current-cost) or last reachable = false) ]
        [
          set pcolor gray
          set reachable replace-item lastID reachable true
          set open lput self open
          set cost replace-item lastID cost current-cost
        ]
        if any? neighbors with [ elev < 0 and last reachable = false ]
        [
          recalculate-cost current-patch
        ]
    ]
  ]
end

to recalculate-cost [border-patch]
  ask border-patch [
    let distance-to-land distance min-one-of borders [distance border-patch]
    ifelse distance-to-land < land-prox
    [
      set pcolor red
      set cost replace-item lastID cost (last cost + (land-prox-weight * (land-prox - distance-to-land) / land-prox ))
      expand-border
    ]
    [
      set pcolor blue
    ]
  ]
end
```

```
to expand-border
  let distance-to-land 0
  ask neighbors with [ pcolor = gray ]
  [
    set distance-to-land distance min-one-of borders [distance myself]
    if distance-to-land < land-prox
    [
      set pcolor red
      set cost replace-item lastID cost (last cost + (land-prox-weight * (land-prox - distance-to-land) / land-prox ))
      expand-border
    ]
  ]
end


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; MOVE ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to go
  place-ships
  while [any? ships][move]
  stop
end

to move
  tick
  let currentPatch 0
  let nextPatch 0
  let layer 0
  ask ships
  [
    set currentPatch patch-here
    if length path <= current-layer [ set current-layer current-layer - 1  ]
    set layer item current-layer path
    ifelse any? anchorages-here with [id = layer]
    [
      set current-layer current-layer + 1
      if length path <= current-layer
      [
        if not landed [ output-show (word "Ship " name " arrived after " (totaltime / 60) " minutes. Fuel consumption: " fuel-cons " tons." ) ]
        set landed true
        let i who
        if one-of [time] of footprints-here with [shipID = i] + (one-of [waittime] of anchorages-here * time-interval) < totaltime
        [
          output-show (word "Ship " name " end to cargo to/from land. " (totaltime / 60) " minutes. ")
          ask anchorages-here [
            set waitlist but-first waitlist
          ]
        die
        ]
      ]
    ]
    [
      let h name
      let next one-of anchorages with [id = layer]

      if first [name] of next = "P"
      [
        ask next [
          if not member? [name] of myself waitlist
          [
            set waitlist lput [name] of myself waitlist
          ]
        ]
        set h first [waitlist] of next
      ]


      if h = name
      [
        set nextPatch min-one-of patches in-cone 2.5 270 with [length reachable > layer and item layer reachable = true] [item layer cost]
        if nextPatch = nobody
        [
          output-show (word "No route found!! ")
          die
          stop
        ]
        face nextPatch
        fd lot-by-timeInterval / miles-per-patch; mpm
        if patch-here != currentPatch
        [
          hatch-footprints 1 [
            set shipID [who] of myself
            set shipspeed [speed] of myself
            set fuelcons [fuel-cons] of myself
            set time [totaltime] of myself
          ]
        ]
      ]
    ]
  set totaltime totaltime + time-interval
  set fuel-cons (((A-fuel-cons * speed * speed)-(B-fuel-cons * speed)) / (24 * 60 * 60)) * totaltime ; ton/seg
  ]
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; EXTRAS ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to-report miles-per-patch
  let world gis:world-envelope ; [ minimum-x maximum-x minimum-y maximum-y ]
  let earth-radius 3959
  report (item 1 world - item 0 world) / (max-pxcor - min-pxcor) * pi / 180 * earth-radius
end

to export-shp
  let abc gis:turtle-dataset footprints
  let filename "NewRoute_"
  set filename (word filename "SS" land-prox "_" "W" land-prox-weight ".shp")
  gis:store-dataset abc filename
end
```

9. Glossary

BFS- breadth first search

A*- A *(star)

ABM- agent based model

SPDHC- sequential point data heatmap comparison

MSRM- Maritime Ship Routing Model

GIS- Geographic Information Systems

AIS- Automatic Identification System

SAIS- Satellite based Automatic Identification System

GPS- Global Positioning System

PMV- Port Metro Vancouver

LP- Land Proximity

LPW- Land Proximity Weight

NN- Nearest Neighbor

SWE- Ship-Wise Errors

CA- Cellular Automata

10. References

Adamatzky, A. (2010). *Game of Life Cellular Automata*: Springer-Verlag London Limited.

Anderson, T. K. (2009). Kernel density estimation and K-means clustering to profile road accident hotspots. *Accident analysis and prevention, 41*(3), 359-364. doi: 10.1016/j.aap.2008.12.014

Ball, H. (2013). *Satellite AIS for Dummies; Special Edition* A. Ahmed (Ed.)

Barbati, M., Bruno, G., & Genovese, A. (2012). Applications of agent-based models for optimization problems: A literature review. *Expert Systems with Applications, 39*(5), 6020-6028. doi: 10.1016/j.eswa.2011.12.015

Beamer, S., Asanovic, K., & Patterson, D. (2012, 10-16 Nov. 2012). *Direction-optimizing Breadth-First Search.* Paper presented at the SC '12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis.

Bialystocki, N., & Konovessis, D. (2016). On the estimation of ship's fuel consumption and speed curve: A statistical approach. *Journal of Ocean Engineering and Science, 1*(2), 157-166. doi: 10.1016/j.joes.2016.02.001

Bianchi, C., Cirillo, P., Gallegati, M., & Vagliasindi, P. A. (2007). Validating and Calibrating Agent-Based Models: A Case Study. *Computational economics, 30*(3), 245-264. doi: 10.1007/s10614-007-9097-z

Blazquez-Soriano, A., & Domenech, J. (2017). Big Data sources and methods for social and economic analyses. *Technological Forecasting and Social Change, 130*. doi: 10.1016/j.techfore.2017.07.027

Bo, C., & Cheng, H. H. (2010). A Review of the Applications of Agent Technology in Traffic and Transportation Systems. *IEEE Transactions on Intelligent Transportation Systems, 11*(2), 485-497. doi: 10.1109/TITS.2010.2048313

Borkowski, P. (2017). The Ship Movement Trajectory Prediction Algorithm Using Navigational Data Fusion. *Sensors, 17*, 1432. doi: 10.3390/s17061432

Bourdon, S., Gauthier, Y., & Greiss, J. (2007). *MATRICS: A Maritime Traffic Simulation*. DRDC – Centre for Operational Research and Analysis

Brake, M. T., Iperen, E. P. A. v., Looije, D., & Koldenhof, Y. (2015). *Unmanned ship simulation with real-time dynamic risk index*.

Clarke, K. C. (2014). Cellular automata and agent-based models. *Handbook of regional science*, 1217-1233.

Crooks, A. (2017). Cellular Automata.

Crooks, A., Castle, C., & Batty, M. (2008). Key Challenges in Agent-Based Modelling for Geo-spatial Simulation. *Computers, Environment and Urban Systems, 32*, 417-430. doi: 10.1016/j.compenvurbsys.2008.09.004

Davidsson, P., Henesey, L., Ramstedt, L., Törnquist, J., & Wernstedt, F. (2005). An analysis of agent-based approaches to transport logistics. *Transportation Research Part C: Emerging Technologies, 13*(4), 255-271. doi: 10.1016/j.trc.2005.07.002

Delling, D., Sanders, P., Schultes, D., & Wagner, D. (2009). Engineering route planning algorithms *Algorithmics of large and complex networks* (pp. 117-139): Springer.

Garg, P. (2022). Breadth first search tutorials & notes: Algorithms. Retrieved March 15, 2022, from https://www.hackerearth.com/practice/algorithms/graphs/breadth-first-search/tutorial/.

Goerlandt, F., & Kujala, P. (2011). Traffic simulation based ship collision probability modeling. *Reliability Engineering & System Safety, 96*(1), 91-107. doi: 10.1016/j.ress.2010.09.003

Graham, M., & Shelton, T. (2013). Geography and the future of big data, big data and the future of geography. *Dialogues in Human Geography, 3*(3), 255-261. doi: 10.1177/2043820613513121

Gunnar Aarsæther, K., & Moan, T. (2009). Estimating Navigation Patterns from AIS. *Journal of Navigation, 62*(04), 587. doi: 10.1017/s0373463309990129

Guo, Y., & Hu, P. (1994). Simulation Model of Transportation System. *IFAC Proceedings Volumes, 27*(12), 513-517. doi: https://doi.org/10.1016/S1474-6670(17)47522-7

Haug, F. S. (2016). Bad Big Data Science. *2016 IEEE INTERNATIONAL CONFERENCE ON BIG DATA (BIG DATA)*, 2863-2871.

Helbing, D., & Balietti, S. (2015). How to Do Agent-Based Simulations in the Future: From Modeling Social Mechanisms to Emergent Phenomena and Interactive Systems Design. *Technical Report 11-06-024*.

Helmreich, S., & Keller, H. (2011). *FREIGHTVISION – Sustainable European Freight Transport 2050: Forecast, Vision and Policy Recommendation*. Verlag Berlin Heidelberg Springer.

Hilliard, R. C., & Pelot, R. (2012). *MATRICS Development and User Reference Guide - Phase 2*. MARIN Risk Lab, Dalhousie University: Defence R&D Canada – CORA.

Holdsworth, J. (1999). The Nature of Breadth-First Search.

Hong Zheng, Y.-J. S., Yi-Chang Chiu, Larry Head, Yiheng Feng, Hui Xi, Sojung Kim, Mark Hickman. (2013). A Primer for Agent-Based Simulation and Modeling in Transportation Applications (pp. 75). University of Arizona, 1401 E University Blvd, Tucson, AZ 85721: University of Arizona.

Huang, J., Cui, Y., Zhang, L., Tong, W., Shi, Y., & Liu, Z. (2022). An Overview of Agent-Based Models for Transport Simulation and Analysis. *Journal of Advanced Transportation, 2022*, 1252534. doi: 10.1155/2022/1252534

Huang, S., Hsu, W., Fang, H., & Song, T. (2016). MTSS -- A Marine Traffic Simulation System and Scenario Studies for a Major Hub Port. *ACM transactions on modeling and computer simulation, 27*(1), 1-26. doi: 10.1145/2897512

Jakob, M., Vaněk, O., & Pěchouček, M. (2011). Using Agents to Improve International Maritime Transport Security. *IEEE Intelligent Systems, 26*(1), 90-96. doi: 10.1109/MIS.2011.23

Kagho, G. O., Balac, M., & Axhausen, K. W. (2020). Agent-Based Models in Transport Planning: Current State, Issues, and Expectations. *Procedia Computer Science, 170*, 726-732. doi: 10.1016/j.procs.2020.03.164

Kanamoto, K., Murong, L., Nakashima, M., & Shibasaki, R. (2020). Can maritime big data be applied to shipping industry analysis? Focussing on commodities and vessel sizes of dry bulk carriers. *Maritime economics & logistics, 23*(2), 211-236. doi: 10.1057/s41278-020-00171-6

Kavak, H., Padilla, J., Lynch, C., & Diallo, S. (2018). *Big Data, Agents, and Machine Learning: Towards a Data-Driven Agent-Based Modeling Approach*.

Kitchin, R. (2013). Big data and human geography. *Dialogues in Human Geography, 3*(3), 262-267. doi: 10.1177/2043820613513388

Kulyk, V., & Sossa, R. (2018). Determining the tourist attractive regions by GIS analysis using heatmaps. *Geodesy and cartography, 44*, 22-27. doi: 10.3846/gac.2018.882

Leigh, R., Louis, S. J., & Miles, C. (2007, 1-5 April 2007). *Using a Genetic Algorithm to Explore A*-like Pathfinding Algorithms.* Paper presented at the 2007 IEEE Symposium on Computational Intelligence and Games.

Li, J., Zhang, X., Yang, B., & Wang, N. (2020). Vessel traffic scheduling optimization for restricted channel in ports. *Computers & Industrial Engineering, 152*, 107014. doi: 10.1016/j.cie.2020.107014

Martineau, E., & Roy, J. (2011). *Maritime Anomaly Detection: Domain Introduction and Review of Selected Literature*. Defence R&D Canada – Valcartier.

Mathew, G. E. (2015). Direction Based Heuristic for Pathfinding in Video Games. *Procedia Computer Science, 47*, 262-271. doi: https://doi.org/10.1016/j.procs.2015.03.206

Mi Yu, A. R. (2015). The Game of life and Other Cellular Automata. Retrieved from Padraic Bartlett Lecture Notes, Math/CCS: Problem-Solving website: https://web.math.ucsb.edu/~padraic/ucsb_2014_15/ccs_problem_solving_w2015/The%20Game%20of%20Life.pdf

Michal Jakob, O. V., Ondrej Hrstka, Michal Pechoucek. (2012). *Agents vs. Pirates: Multi-Agent Simulation and Optimization to Fight Maritime Piracy*. Paper presented at the 12th International Conference on Autonomous Agents and Multiagent Systems.

Mihailescu, C. (2019). Pathfinding Visualizer. from https://clementmihailescu.github.io/Pathfinding-Visualizer/

Miller, A. (2019). The intrinsically linked future for human and Artificial Intelligence interaction. *Journal of Big Data, 6*(1), 38. doi: 10.1186/s40537-019-0202-7

Miller, H. J., & Goodchild, M. F. (2014). Data-driven geography. *GeoJournal, 80*(4), 449-461. doi: 10.1007/s10708-014-9602-6

Mirovic, M., Milicevic, M., & Obradovic, I. (2018). Big Data in the Maritime Industry. *Nase More, 65*. doi: 10.17818/NM/2018/1.8

Netek, R., Pour, T., & Slezakova, R. (2018). Implementation of Heat Maps in Geographical Information System – Exploratory Study on Traffic Accident Data. *Open Geosciences, 10*(1), 367-384. doi: 10.1515/geo-2018-0029

Norris, A. (2006). AIS Implementation – Success or Failure? *Journal of Navigation, 60*(01), 1. doi: 10.1017/s0373463307004031

Ondrej Vanek, M. J., Michal Pechoucek. (2013). Using Data-Driven Simulation for Analysis of Maritime Piracy. In E. Bosse (Ed.), *Prediction and Recognition of Piracy Efforts Using Collaborative Human-Centric Information Systems* (pp. 109-116): IOS Press.

Ondrej Vanek, O. H., Michal Jakob, Michal Pechoucek. (2011). AgentC: Agent-based System for Securing Maritime Transit (Demonstration). *Proceedings of The 10th International Conference on Autonomous Agents and Multiagent Systems*.

Ondrej Vanek, O. H., Michal Pechoucek. (2014). Improving Group Transit Schemes to Minimize Negative Effects of Maritime Piracy. *IEEE Intelligent Transportation Systems*.

Perera, L., Mo, B., & Kristjánsson, L. (2015). *Identification of Optimal Trim Configurations to improve Energy Efficiency in Ships* (Vol. 48).

. *Port Metro Reference Guide*. (2016). Harbour Master Port Metro Vancouver: Llyod's MIU.

Rachmawati, D., & Gustin, L. (2020). Analysis of Dijkstra's Algorithm and A* Algorithm in Shortest Path Problem. *Journal of Physics: Conference Series, 1566*, 012061. doi: 10.1088/1742-6596/1566/1/012061

Railsback, S., Lytinen, S., & Jackson, S. (2006). Agent-based Simulation Platforms: Review and Development Recommendations. *SIMULATION, 82*, 609-623. doi: 10.1177/0037549706073695

Report to Canadians: Investing in our coasts through the Oceans Protection Plan. (2021). Retrieved 03 - 11, 2022, from https://tc.canada.ca/en/initiatives/oceans-protection-plan/report-canadians-investing-our-coasts-through-oceans-protection-plan#preserving

Safaei, A. A., Ghassemi, H., & Ghiasi, M. (2019). VLCC's fuel consumption prediction modeling based on noon report and automatic identification system. *Cogent Engineering, 6*(1), 1595292. doi: 10.1080/23311916.2019.1595292

Sarkar, P. (2000). A Brief History of Cellular Automata. *ACM Comput. Surv., 32*, 80-107. doi: 10.1145/349194.349202

Scheutz, M., & Mayer, T. (2016). Combining Agent-Based Modeling with Big Data Methods to Support Architectural and Urban Design (pp. 15-31).

Serenko, A., & Detlor, B. (2003). Agent toolkit satisfaction and use in higher education. *Journal of Computing in Higher Education, 15*, 65-88. doi: 10.1007/BF02940853

Słomska-Przech, K., Panecki, T., & Pokojski, W. (2021). Heat Maps: Perfect Maps for Quick Reading? Comparing Usability of Heat Maps with Different Levels of Generalization. *ISPRS International Journal of Geo-Information, 10*(8). doi: 10.3390/ijgi10080562

Smith, T., O'Keeffe, E., Aldous, L., & Agnolucci, P. (2013). *Assessment of Shipping's Efficiency Using Satellite AIS data*

Stevens, G. (2019). Introduction to agent-based models and cellular automata. DEV Community. Retrieved March 2, 2022, from https://dev.to/thegeoffstevens/introduction-to-agent-based-models-and-cellular-automata-jnf.

Sui, D., Goodchild, M., & Elwood, S. (2012). Volunteered Geographic Information, the Exaflood, and the Growing Digital Divide (pp. 1-12). Dordrecht: Springer Netherlands.

Swift, N. (2020). Easy A* (star) pathfinding. Medium. Retrieved March 2, 2022, from https://medium.com/@nicholas.w.swift/easy-a-star-pathfinding-7e6689c7f7b2.

Tam, C., Bucknall, R., & Greig, A. (2009). Review of Collision Avoidance and Path Planning Methods for Ships in Close Range Encounters. *Journal of Navigation, 62*(03), 455. doi: 10.1017/s0373463308005134

Tesfatsion, L., & Judd, K. L. (2006). *Handbook of Computational Economics: Agent-Based Computational Economics*: Elsevier Science.

Tisue, S. (2004). *NetLogo: Design and implementation of a multi-agent modeling environment*.

Tixerant, M., Guyader, D., Gourmelon, F., & Queffelec, B. (2018). How can Automatic Identification System (AIS) data be used for maritime spatial planning? *Ocean & Coastal Management, 166*. doi: 10.1016/j.ocecoaman.2018.05.005

Torre-Bastida, A. I., Del Ser, J., Laña, I., Ilardia, M., Bilbao, M. N., & Campos-Cordobés, S. (2018). Big Data for transportation and mobility: recent advances, trends and challenges. *IET Intelligent Transport Systems, 12*(8), 742-755.

Vanek, O. (2013). *Dynamic Group Transit Scheme for Corridor Transit*. Paper presented at the Proceedings of the 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO).

Vaněk, O., Jakob, M., Hrstka, O., & Pěchouček, M. (2013). Agent-based model of maritime traffic in piracy-affected waters. *Transportation Research Part C, 36*(Complete), 157-176. doi: 10.1016/j.trc.2013.08.009

Vlahogianni, E. (2015). Computational Intelligence and Optimization for Transportation Big Data: Challenges and Opportunities (pp. 107-128).

Vouros, G., Doulkeridis, C., Santipantakis, G., & Vlachou, A. (2018). Taming Big Maritime Data to Support Analytics (pp. 15-27).

Wu, J., Peng, H., Ohtsu, K., Kitagawa, G., & Itoh, T. (2012). Ship's tracking control based on nonlinear time series model. *Applied Ocean Research, 36*, 1-11. doi: 10.1016/j.apor.2012.01.004

Yu, E. (2002). Agent-Oriented Modelling: Software Versus the World. *Faculty of Infromation Sciences*.

Zhou, Y., Daamen, W., Vellinga, T., & Hoogendoorn, S. (2019). Review of maritime traffic models from vessel behavior modeling perspective. *Transportation research. Part C, Emerging technologies, 105*, 323-345. doi: 10.1016/j.trc.2019.06.004