Coordinating Vessel Traffic to Improve Safety and Efficiency

Teck-Hou Teng
Urban Computing and Engineering
Corporate Laboratory
71 Stamford Road
Singapore 178895
thteng@smu.edu.sg

Hoong Chuin Lau
School of Information Systems
80 Stamford Road
Singapore 178902
hclau@smu.edu.sg

Akshat Kumar
School of Information Systems
80 Stamford Road
Singapore 178902
akshatkumar@smu.edu.sg

ABSTRACT

Global increase in trade leads to congestion of maritime traffic at the ports. This often leads to increased maritime incidents or near-miss situations. To improve maritime safety while maintaining efficiency, movement of vessels needs to be better coordinated. Our work formulates this problem of coordinating the paths of vessels as a multi-agent path-finding (MAPF) problem. To address this problem, we introduce an innovative application of MAPF in the maritime domain known as Vessel Coordination Module (VCM). Based on the local search paradigm, VCM plans on a joint state space updated using the Electronic Navigation Charts (ENC) and the paths of vessels. We introduce the notion of path quality that measures the number of positions on a vessel path that is too close to some other vessels spatially and temporally. VCM aims to improve the overall path quality of vessels by improving path quality of selected vessels. Experiments are conducted on the Singapore Straits to evaluate and compare performance of our proposed approach in heterogeneous maritime scenario. Our experiment results show that VCM can improve the overall path quality of the vessels.

Keywords

Multi-agent path finding; coordination; vessel traffic management

1. INTRODUCTION

The Straits of Malacca has long been a major passageway for movement of goods to and from Asia. As the busiest transshipment port in the world, Singapore faces challenges of receiving larger vessels and increased volume of vessel traffic calling on her port as well as in transition. Due to environment and regulatory constraints, vessels can only use narrow strips of water passages for entering and leaving the ports. Congestion along certain parts of Singapore Strait leads to increased risk of maritime incidents and near-miss situations. To mitigate congestion for improving safety and efficiency of vessel traffic, the movement of vessels needs to be better coordinated. This task of coordinating vessel traffic is made possible by the wide adoption of the standard Automatic Identification System (AIS) on vessels and by Vessel Traffic Services (VTS). It is possible to make use of AIS data to determine paths of vessels, and apply intelligent technology to coordinate vessel traffic.

Recent years saw gradual adoption of multi-agent technologies by the industry [12]. In particular, agent-based simulation is used for addressing problems in international maritime transport security [6]. Agent behaviors are represented using finite-state machines. Vessel operational characteristics and environmental data are included to improve the realism of the simulation. Individual plans and coordination methods can then be evaluated reliably. VTS domain is a complex socio-technical system [13]. Control over vessel traffic is maintained through a mixture of opportunistic and tactical control strategies. More strategic and tactical control within daily VTS operations are recommended. Existing literature in VTS domain are mostly descriptive. Their focus is mainly on developing tools for aggregating multiple streams of information to aid decision-making. Techniques recommending courses-of-actions are rare.

Our work addresses a real world problem of coordinating movement of vessels by considering dynamic changes happening at sea. A key contribution of our work lies in formulating this problem as a multi-agent path-finding (MAPF) problem contextualized with the additional complexity of maritime traffic and port operation activities. To solve this problem, we introduce an innovative application of search algorithm in maritime domain known as Vessel Coordination Module (VCM). VCM coordinates movement of heterogeneous vessels (agents) by replanning paths of selected agents to mitigate congestion and near-miss situations. Planning on a joint state space, the cost of nodes is based on Electronic Navigation Chart features and the known paths of agents. We define the notion of path quality that measures the number of positions on an agent path which is too close to some other vessels spatially and temporally.

As a second contribution of this paper, we proposed a Coordinated Path Finding (CPF) algorithm used within VCM to find paths with the best aggregated path quality. Based on the local search paradigm, variants of the proposed CPF algorithm are evaluated. Experiments are conducted to evaluate the efficacy of CPF algorithm in heterogeneous maritime scenarios based on maritime data of Singapore Straits. The performance of several variants of VCM are compared to identify the best performing variant. Our experimental results show the best performing variant of CPF algorithm is capable of improving the aggregated path quality.

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

The presentation of this work continues in Section 2 with the related works. We motivate and define the problem in Section 3. We present our proposed Vessel Coordination Module (VCM) in Section 4. The experiments and the results are presented in Section 5. Conclusion and a summary of our work is presented in Section 6.

2. RELATED WORK

Related works based on the topic of vessel traffic management are surveyed in Section 2.1. Related works on multi-agent path-finding are surveyed in Section 2.2.

2.1 Vessel Traffic Management

Several recent works on the topic of vessel traffic management are known. Most of these works aim to improve operational efficiency and safety of vessels. The survey focus on their techniques for managing vessel traffic.

A Vessel Traffic Management and Information System (VTMIS) developed in [4] has an expert system comprising vessel motion models, hydro-dynamic prediction model and vessel traffic flow model. The use of that VTMIS is aimed at preventing or reducing maritime incidents occurring in narrow waterways such as the Turkish Straits. There is also a vessel traffic management system (VTMS) for cooperative decision-making among all actors of vessel traffic management [10]. That VTMS derives vessel paths by accounting for possible drift, meteorological conditions and sea state to improve situation-awareness for all actors. Decision aids are provided to help the actors to find solution to their problem in heterogeneous maritime environment.

A decision support system comprising a spatio-temporal geo-referenced database, a simulator and data post-processing tools is developed in [11]. It is used for managing the navigation activities in the St. Lawrence River Estuary in Canada. Paths of vessels are based on AIS data of real vessel traffic. The simulator is used for testing management scenarios of vessel traffic. Another decision support system based on Artificial Neural Network (ANN) is developed in [14]. ANN is trained using real data from Automatic Identification System (AIS). The trained ANN is trained to predict position of vessels three minutes from the initial point of evaluation. The predicted positions inform VTS operators of evolving conditions of vessel traffic. The routing of ships for the entire voyage can also be formulated as a mixed-integer programming model and solved using rolling horizon heuristic (RHH) [1]. RHH integrates speed optimization and planning of shipping paths. It may also be suitable for problems with long planning horizon.

Almost all the above-mentioned works focus on aggregating data and information from multiple sources to support decision-making of users. There is little or no attempt to coordinate movement of vessels by prescribing paths. Understandably, it is non-trivial to prescribe alternative paths due to lack of effective coordination strategy for the maritime domain where there is the need to consider existing regulatory and environmental constraints. We hope to address this issue by introducing an application capable of prescribing *safer* and *more effective* paths to vessels.

2.2 Multi-Agent Path Finding

Multi-agent path finding (MAPF) problem is a well-studied problem. Numerous variants of MAPF problem are known. Several recent works addressing some of these MAPF problems are surveyed here.

A high performance path-planning algorithm for 3D+time planning known as *Accelerated A** is developed in [15]. Performance of Accelerated A* algorithm is compared with Theta*, Rapid-Exploring Random Trees and original A* algorithms. Experiment results show Accelerated A* can give the shortest paths. But it is slower than Theta* and Rapid-Exploring Random Trees. Accelerated A* is parallelized in [8] to accelerate planning of path. A hashing function based on geographical partitioning of search space is used to schedule the workload at the cores. Experiment results reveal significant reduction of planning time while preserving the quality of the planned paths.

The works developed in [2] and [7] are most similar to our work. A penalty-based method known as k-step Penalty Method (kPM) for improving success rate and quality of planned collision-free path is seen in [2]. Similar to [2], our work coordinates movement of agent i by considering relevant information of neighboring agent j when updating joint state space.

An anytime algorithm known as ORCA-RRT* integrating a reactive approach known as ORCA and a sampling-based algorithm known as RRT* algorithm is developed in [7]. ORCA-RRT* algorithm patches weaknesses of either algorithms for finding coordinated paths for holonomic agents in 2-D polygonal environments. Similar to [7], our work propose solution for finding coordinated paths of agents with unsafe paths. We are using a joint state space for finding paths that allow vessels to move in coordinated manner without the need for any explicit coordination among them.

In essence, our work addressed MAPF problem with added complexity of maritime domain. A multi-agent system in maritime domain is distributed and heterogeneous. Vessels, represented as agents, have different movement characteristics, dimensions and ship domains. In contrast to many MAPF problems, our work considers agent i occupies multiple nodes on a joint state space $\mathcal{G}$. The size of the nodes is dependent on the minimum SOG v_{min} of vessels in area-of-interest (AOI). Like many other MAPF problems, our nodes have the same dimension. But, our work accounts for agents moving at different speeds along its path.

3. PROBLEM DEFINITION

This section describes a motivating problem in Section 3.1 and defines the problem formally in Section 3.2.

3.1 Motivating Problem

Existing vessel traffic regulation requires vessels to enter and leave port waters using designated regions of the traffic separation scheme (TSS) [16]. One such designated region in Singapore Straits is illustrated using Fig. 1. It is non-trivial to plan incident-free paths without coordinating movement of vessels. Typically, this task of coordinating movement of vessels is taken up by Vessel Traffic Service (VTS). Thus, we are introducing an innovative application capable of suggesting coordinated paths to vessels with *unsafe* paths.

3.2 Problem Statement

In the maritime domain, it is necessary to consider many environmental and regulatory features when planning paths for vessels. This work addresses a multi-agent path finding (MAPF) problem defined with a population $\mathcal{N}$ of agents

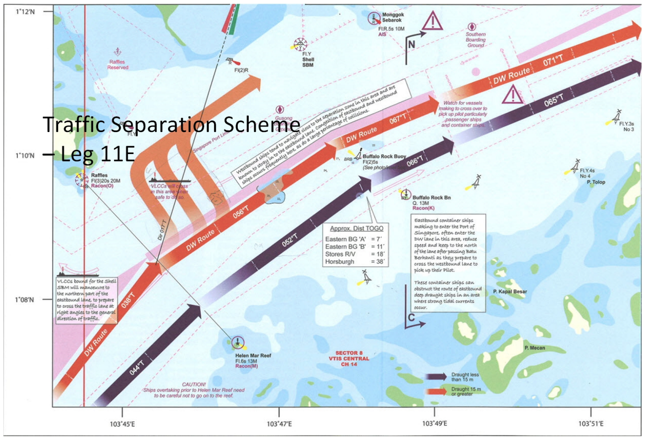


Figure 1: A designated region of the Traffic Separation Scheme of Singapore Straits where vessels use to enter or leave port waters of Singapore.

operating on joint state space $\mathcal{G}$.

Definition 3.1. *Joint State Space $\mathcal{G} = \{N(\mathcal{G}), E(\mathcal{G})\}$ is a graph comprising a set of nodes $N(\mathcal{G})$ and edges $E(\mathcal{G})$ defined for an area-of-interest (AOI).*

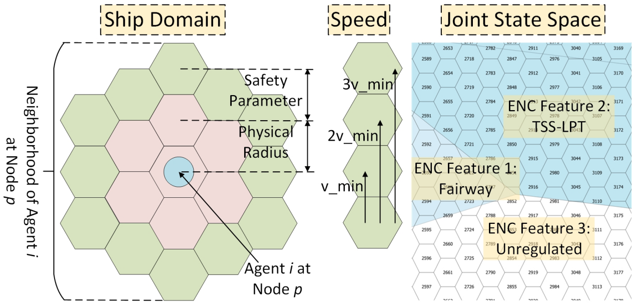


Figure 2: Left: An illustration of neighborhood $\mathcal{T}_N(p)$ at Node p. Middle: An illustration of cruising speed-over-ground (SOG) of agent expressed as multiples of v_{min}. Right: Illustration of three types of ENC-based hexagon grids in joint state space $\mathcal{G}$.

As illustrated in Fig. 2(right), node $p \in \mathcal{G}$ is a hexagon. Therefore, node p has a maximum degree of 6. Edge $e_{(p,q)} \in E(\mathcal{G})$ connects adjacent nodes p and q. Distance $d(p,q)$ between adjacent nodes p and q is denoted as $d_{min}(p,q)$. Distance $d(p,r)$ between non-adjacent nodes p and r is specified as multiples of $d_{min}(p,q)$. The minimum distance $d_{min}(p,q)$ is derived using

$$d_{min}(p,q) = v_{min} \times \Delta t \tag{1}$$

where v_{min} is the minimum cruising SOG of vessels and Δt is the sampling interval of agent positions.

Definition 3.2. *Agent $i \in \mathcal{N}$ represents a vessel having a circular ship domain defined by its* physical radius *r_i and* safety perimeter *s_i, and a path $\pi_i(o_i, d_i, v_i)$ specifying its movement from node o_i to node d_i at cruising SOG v_i.*

For simplicity, we assume that each agent starts its path from o_i at time step 0. The *cruising SOG v_i* of agent i determines the number of grids it will move in one time step. It lies in range defined using v_{min} and v_{max}. As illustrated in Fig. 2(middle), it is expressed as multiples of v_{min}. For each agent i, if $v_i = v_{min}$, then the two consecutive nodes on its path are adjacent, otherwise, they are not adjacent.

The *physical radius r_i* of agent i is the circular region of grids emanating from the centroid grid. As illustrated in Fig. 2(left), r_i is based on vessel type. The *safety perimeter s_i* is the additional number of circular region of grids beyond r_i. This is the distance agent i must keep from agent j in order to be deemed safe. As illustrated in Fig. 2(left), the *safety perimeter* may vary among vessels of the same type.

As defined in Definition 3.2, agent i has a circular ship domain. Given an agent i on a node k at time t, we can define its neighborhood $\mathcal{T}(k)$ as follows.

Definition 3.3. *Neighborhood $\mathcal{T}(k)$ of agent i at node k is a layered subgraph of $\mathcal{G}$ with node k as the only node at layer 0 and all its adjacent nodes at layer 1, nodes adjacent to layer 1's nodes at layer 2, and so on, ending at layer $r_i + s_i - 1$.*

The safety level of path π_i of agent i is determined with respect to its neighborhood $\mathcal{T}(k)$ at each location k along its path. Let k and l denote the respective locations of agents i and j respectively at time t. Then,

Definition 3.4. *An Unsafe Condition exists when $\mathcal{T}(k) \cap \mathcal{T}(l) \neq \emptyset$.*

Hence, to prevent the unsafe condition from occurring, it is necessary to coordinate movement of agent i and agent j such that at every time step t, we require that $\{\mathcal{T}(k) \cap \mathcal{T}(l) \equiv \emptyset\}$ where k and l denote agents i and j's locations respectively at time t.

4. COORDINATING VESSEL TRAFFIC

Here, we introduce our innovative application of search algorithm in maritime domain known as the Vessel Coordination Module (VCM) in this section. Given a maritime scenario of vessels in an area of interest (AOI), the goal of VCM is to recommend coordinated paths for the vessels. Inputs to VCM are described in Section 4.1. A search algorithm known as Coordinated Path Finding (CPF) algorithm is described in Section 4.2. Outputs of VCM are described in Section 4.3.

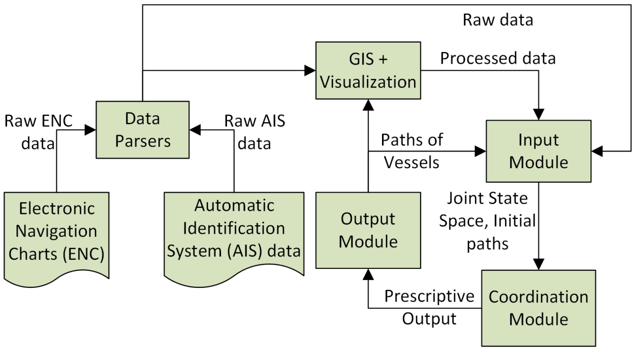


Figure 3: Architecture of Vessels Coordination Module (VCM).

The architecture of VCM is illustrated in Fig. 3. Capable of operating in either *historical* or *live* mode, VCM accepts data from Automatic Identification System (AIS) and Electronic Navigation Chart (ENC) as raw inputs. The raw inputs are processed and visualized using a Geographic Information Software (GIS) and forwarded to the Input Module. The Input Module transforms the raw and processed data into the joint state space and the initial paths for the Search Module. Using these inputs, the Search Module searches for the coordinated paths. The Output Module determines the aggregated path quality, organizes the paths and sends it to GIS visualization tool and as feedback to the Input Module.

4.1 Input

From Fig. 3, the Coordination Module of VCM uses the initial paths $\Pi(\mathcal{N}, 0)$ of agent population $\mathcal{N}$, the initial joint state space $\mathcal{G}(0)$ and the selected ENC features as inputs.

Agent Paths $\Pi(\mathcal{N})$: At a slight abuse of notation, we let $\Pi(\mathcal{N}, t)$ denote the set of paths associated with agents $\mathcal{N}$ at iteration t, and $\mathcal{Q}(\Pi(\mathcal{N}, t))$ denote the aggregated path quality of $\Pi(\mathcal{N}, t)$. Given $\Pi(\mathcal{N}, 0)$, VCM seeks to improve $\mathcal{Q}(\Pi(\mathcal{N}, t))$ by improving path quality of selected agent paths at iteration t. More precisely, VCM generates the set of paths at the final iteration T_s such that $\mathcal{Q}(\Pi(\mathcal{N}, T_s)) > \mathcal{Q}(\Pi(\mathcal{N}, 0))$.

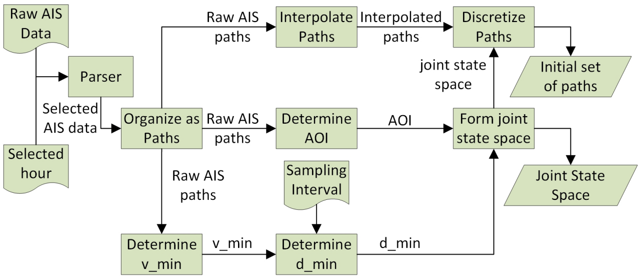


Figure 4: Illustration of processes for extracting the initial paths $\Pi(\mathcal{N}, 0)$ from AIS data and forming the joint state space $\mathcal{G}(0)$.

When VCM is used in *historical* mode, VCM operates on *historical* AIS data. Initial paths $\Pi(\mathcal{N}, 0)$ comprise paths extracted from AIS data of vessels operating in selected AOI. From Fig. 4, AIS data from the selected time frame is rearranged to form the paths of vessels. An area-of-interest is identified by analyzing paths of vessels for areas of high vessel traffic volume with significant concern for safety and efficiency of vessels. After that, paths of vessels are trimmed to only contain positions within the AOI. Positions on paths of vessels are interpolated to make sure positions of all vessels are known at the same time marks. After that, initial paths $\Pi(\mathcal{N}, 0)$ is formed using initial joint state space $\mathcal{G}(0)$ by translating positions on paths specified in GPS coordinates into identification (ID) number of nodes on $\mathcal{G}(0)$.

When VCM is used in *live* mode, VCM operates on *live* data from the StraitRep of vessels. The initial paths $\Pi(\mathcal{N}, 0)$ are predicted using the initial joint state space $\mathcal{G}(0)$ and origin and destination (O-D) pairs of vessels. Such information is available because it is mandatory for vessels navigating in Straits of Malacca and Singapore (SOMS) to submit StraitRep at regular interval [9]. O-D pairs of vessels from the StraitRep are specified in GPS coordinates and have to be translated into node ID using $\mathcal{G}(0)$. After that, $\Pi(\mathcal{N}, 0)$ can be predicted using the translated O-D pairs.

Joint State Space $\mathcal{G}$: Agents use joint state space $\mathcal{G}$ for path finding. As defined in Definition 3.1, $\mathcal{G}$ is defined for an area-of-interest (AOI). Size of grid is defined using (1). Also illustrated in Fig. 4, minimum SOG v_{min} is determined by analyzing the raw paths of vessels in AOI. Sampling interval Δt is an user choice. Using d_{min}, joint state space $\mathcal{G}(0)$ is formed as a collection of hexagon nodes of equal length at all sides.

Each node $i \in N(\mathcal{G})$ has cost feature $\mathcal{C}_i$ such that $\mathcal{C}_i = \frac{\alpha_i(t)+\beta_i}{2}$ where $\alpha_i(t)$ is the aggregated cost of time-dependent cost features and β_i is the aggregated cost of time-independent cost features. The time-dependent cost features of node i are based on intersection check performed on the neighborhood of all nodes on path of all agents. The idea is to increase the cost of nodes common to the neighborhoods of nodes on paths of different agents at the same iteration.

Algorithm 1 Update Joint State Space $\mathcal{G}$.

```
Require: Set of paths $\Pi(\mathcal{N}, t_s)$ and joint state space $\mathcal{G}(t_s)$
for agent $i \in \mathcal{N}$ do
  for node $k \in \pi_i$ do
    Form neighborhood $\mathcal{T}(k)$ with node $k$ as the layer 0 node
    Get position of node $k$ in $\pi_i$ as time step $t$
    for agent $j \in \mathcal{N}$ and $i \neq j$ do
      Get position of agent $j$ at time step $t$ as node $l$
      Form neighborhood $\mathcal{T}(l)$ with node $l$ as the layer 0 node
      if $\mathcal{T}(l) \cap \mathcal{T}(k) \neq \emptyset$ then
        Calculate $\alpha_m(t)$ of node $m \in \mathcal{T}(k)$ w.r.t. node $n \in \mathcal{T}(l)$ using
            $\alpha_m^{new}(t_s) = \frac{\alpha_m^{old}(t_s) + \frac{\delta_{max} - \|m-n\|}{\delta_{max}}}{2}$   (2)
        where $\delta_{max}$ is derived using
            $\delta_{max} = \max_{r \in \mathcal{T}(l), s \in \mathcal{T}(k)} \|r - s\|$   (3)
        Update cost $\mathcal{C}_m$ of node $m$ using $\frac{\alpha_m^{new}+\beta_m}{2}$
      end if
    end for
  end for
end for
return updated joint state space $\mathcal{G}(t_s)$
```

The process of updating $\mathcal{C}_i$ is outlined in Algorithm 1 and described as follows.

Consider two agents i and j having paths π_i and π_j respectively. At iteration t, agent i is at node $k \in \pi_i$ and agent j is at node $l \in \pi_j$. Node k has neighborhood $\mathcal{T}(k)$ and node l has neighborhood $\mathcal{T}(l)$ such that $\mathcal{T}(l) \cap \mathcal{T}(k) \neq \emptyset$. Intersection between $\mathcal{T}(k)$ and $\mathcal{T}(l)$ at iteration t triggers update of $\alpha_m^{new}(t_s)$ of node k using (2).

ENC Features: ENC is used by vessels to navigate the waterways. Normally used with a Electronic Chart Display and Information System (ECDIS), selected ENC features are used here for calculating aggregated cost of time-independent cost features β_m.

Our work uses a fixed number N_{ENC} of ENC features for calculating β_m of node m. Our choice of ENC features are shown in Table 2. The ENC features are scattered throughout the AOI. Therefore, node $m \in N(\mathcal{G})$ will not have all ENC features. A specific cost $\beta_m(ENC_e)$ is attributed for ENC feature ENC_e in node m. The normalized time-independent cost β_m is derived using

$$\beta_m = \frac{\sum_e^{N_{ENC}} \beta_m(ENC_e)}{N_{ENC}} \tag{4}$$

Following that, standard path planner such as $A*$ [3] may be used to construct paths avoiding the *expensive* nodes, thereby achieving the effect of coordinating movement of agents.

4.2 The Search Strategy

The strategy for coordinating movement of vessels known as Coordinated Path Finding (CPF) algorithm is outlined in Algorithm 2. Based on the local search paradigm, CPF algorithm searches for a solution comprising a set of paths $\Pi(\mathcal{N})$ such that $\mathcal{Q}(\Pi(\mathcal{N}, t_s))$ approaches 1.0. An optimal solution has $\mathcal{Q}(\Pi(\mathcal{N}, t_s)) \equiv 1.0$ indicating none of the paths in $\Pi(\mathcal{N}, t_s)$ is *unsafe* (see Definition 3.4).

CPF begins search ($t_s = 0$) from an initial set of paths $\Pi(\mathcal{N}, 0)$ and an initial joint state space $\mathcal{G}(0)$. *State* of outer loop is a solution point $s_p(t_s)$ of the search process. A solution point $s_p(t_s)$ at search iteration t_s has a set of path $\Pi(\mathcal{N}, t_s)$ whose aggregated path quality $\mathcal{Q}(\Pi(\mathcal{N}, t_s))$ lies in range $0 \leq \mathcal{Q}(\Pi(\mathcal{N}, t_s)) \leq 1.0$.

At the inner loop, a neighborhood search at solution point $s_p(t_s)$ begins with all agents in open list Θ_o. At search iteration t_s, agent i is selected from Θ_o using a heuristic strategy known as α-strategy. A standard path planning such as A^* [3] is used to plan a trial path $\pi_i^t(t_s)$ on $\mathcal{G}(t_s)$ for agent i. Aggregated path quality $\mathcal{Q}(\Pi(\mathcal{N}, t_s))$ is determined using trial path $\pi_i^t(t_s)$. Agent i is added to winner list Θ_w when $\mathcal{Q}(\Pi(\mathcal{N}, t_s))$ is better than the best-so-far aggregated path quality $\mathcal{Q}^*(\Pi(\mathcal{N}))$. Neighborhood search at $s_p(t_s)$ continues until $\Theta_o \equiv \emptyset$.

Algorithm 2 Coordinated Path Finding (CPF) Algorithm.

```
Require: Initial set of paths $\Pi(\mathcal{N}, 0)$ and initial joint state space $\mathcal{G}(0)$
Determine initial aggregated path quality $\mathcal{Q}(\Pi(\mathcal{N}, 0))$
At $t_s = 0$, initialize joint state space $\mathcal{G}(t_s)$ using $\Pi(\mathcal{N}, 0)$
while $\mathcal{Q}(\Pi(\mathcal{N}, t_s - 1)) \neq \mathcal{Q}(\Pi(\mathcal{N}, t_s))$ do
    At solution point $s_p(t_s)$, populate open list $\Theta_o$ with all agents
    while ($\Theta_o \neq \emptyset$) do
        Select agent $i$ using $\alpha$-strategy from open list $\Theta_o$
        Plan trial path $\pi_i^t$ using joint state space $\mathcal{G}(t_s)$
        Update joint state space as $\mathcal{G}^t(t_s)$ using $\pi_i^t$
        Determine aggregated path quality $\mathcal{Q}^t(\Pi(\mathcal{N}, t_s))$ using $\mathcal{G}^t(t_s)$
        if Better($\mathcal{Q}^t(\Pi(\mathcal{N}, t_s)), \mathcal{Q}^*(\Pi(\mathcal{N}))$) then
            $\mathcal{Q}^*(\Pi(\mathcal{N})) = \mathcal{Q}^t(\Pi(\mathcal{N}, t_s))$
            Add agent $i$ to winner list $\Theta_w$
            Save $\pi_i^t$ as candidate path $\pi_i^c$
        end if
        Revert to path $\pi_i$ for agent $i$ and $\mathcal{G}(t_s)$
        Transfer agent $i$ from open list $\Theta_o$ to closed list $\Theta_c$
    end while
    Select winning agent $i^*$ using $\beta$-strategy from winner list $\Theta_w$
    Commit $\pi_i = \pi_i^c$ to create new solution point $s_p^*$
    Update joint state space $\mathcal{G}(t_s)$ using $\pi_i$
    Clear closed list $\Theta_c$ and winner list $\Theta_w$.
    $t_s^{new} = t_s^{old} + 1$ and $s_p(t_s^{new}) = s_p^*(t_s^{old})$
end while
return coordinated path $\Pi(\mathcal{N}, t_s)$
```

After neighborhood search, winning agent i^* is selected from Θ_w using another heuristic strategy known as β-strategy. The saved candidate path $\pi_i^c(t_s)$ become path $\pi_i(t_s)$ of agent i, creating new solution point $s_p^*(t_s)$. Joint state space $\mathcal{G}(t_s)$ is then updated for s_p^*. Closed list Θ_c and winner list Θ_w are cleared for the next search iteration. The search continues until a terminal solution point $s_p(t_s)$ where $\mathcal{Q}(\Pi(\mathcal{N}, t_s - 1)) \equiv \mathcal{Q}(\Pi(\mathcal{N}, t_s))$.

4.3 Output

VCM produces three kinds of outputs at the final search iteration T_s. The aggregated path quality $\mathcal{Q}(\Pi(\mathcal{N}, T_s))$ is the quantitative output. As outlined in Algorithm 3, path quality $q(\pi_i)$ is calculated using a similar approach of checking intersection between neighborhoods of nodes on paths π_i and π_j. To determine $q(\pi_i)$, it is sufficient to track the number of intersection π_i has with some other paths using $\omega(\pi_i)$. After that, $q(\pi_i)$ is calculated using (5).

Algorithm 3 Calculate Path Quality $q(\pi_i)$ of Agent i.

```
Require: Path $\pi_i$ of agent $i$ leading from Node $o_i$ to Node $d_i$
Ensure: $\omega(\pi_i) = 0$
for node $k \in \pi_i$ do
    Form neighborhood $\mathcal{T}(k)$ with node $k$ as the layer 0 node
    Get position of node $k$ in $\pi_i$ as time step $t$
    for agent $j \in \mathcal{N}$ and $i \neq j$ do
        Get position of agent $j$ at time step $t$ as node $l$
        Form neighborhood $\mathcal{T}(l)$ with node $l$ as the layer 0 node
        if $\mathcal{T}(k) \cap \mathcal{T}(l) \neq \emptyset$ then
            $\omega^{new}(\pi_i) = \omega^{old}(\pi_i) + 1$
        end if
    end for
    Calculate $q(\pi_i)$ using

        $q(\pi_i) = 1.0 - \frac{\omega(\pi_i)}{|\pi_i|}$        (5)

end for
return path quality $q(\pi_i)$
```

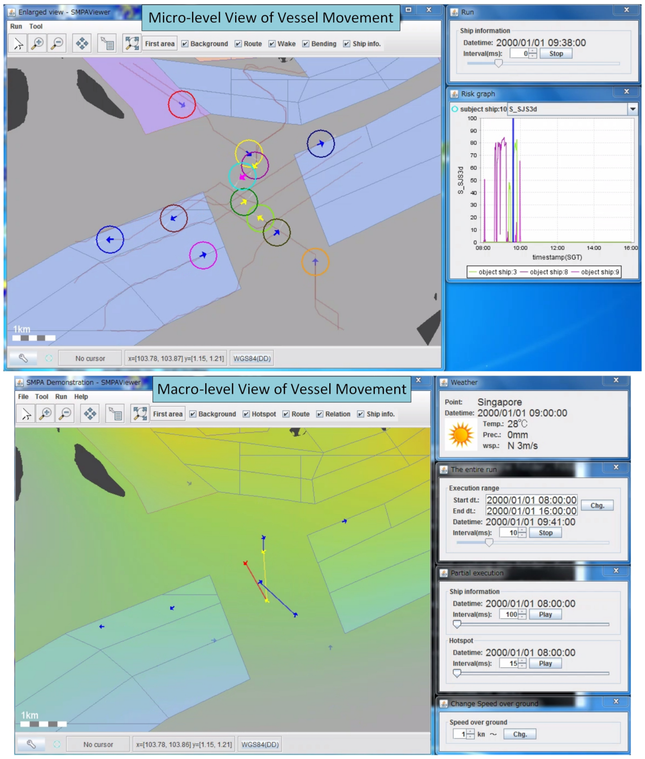
There is also the set of paths $\Pi(\mathcal{N}, T_s)$ discovered using CPF algorithm. For $\Pi(\mathcal{N}, T_s)$ to be usable, ID of node k on path π_i has to be translated into GPS coordinates by using the same mapping function for translating from GPS coordinates to node ID in reversed order. Only after that, paths $\Pi(\mathcal{N}, T_s)$ can be used to guide vessels to navigate in a coordinated manner without any explicit coordination among them. As illustrated in Fig. 5, the translated $\Pi(\mathcal{N}, T_s)$ can also be visualized using a GIS visualization tool [17].

5. PERFORMANCE EVALUATION

Experiments are conducted to evaluate the efficacy of VCM in addressing the MAPF problem of coordinating multiple vessels in heterogeneous maritime domain. The design choices of the experiments are presented in Section 5.1. The experiment results are presented in Section 5.2.

5.1 Design of Experiments

Experiments are conducted for evaluating performance of VCM on selected key parameters comprising safety parameters λ, α-strategies, β-strategies and search criteria s_c. Safety parameters λ used in the experiments are $\lambda = \{1, 2, 3, 4\}$.



No cursor | x=[103.78, 103.87] y=[1.15, 1.21] | WGS84(DD)

Ship information
Datetime: 2000/01/01 09:38:00
Interval(ms): 0 Stop

subject ship:10 S_SJS3d

No cursor | x=[103.78, 103.86] y=[1.15, 1.21] | WGS84(DD)

Point: Singapore
Datetime: 2000/01/01 09:00:00
Temp.: 28℃
Prec.: 0mm
wsp.: N 3m/s

Execution range
Start dt.: 2000/01/01 08:00:00
End dt.: 2000/01/01 16:00:00 Chg.
Datetime: 2000/01/01 09:41:00
Interval(ms): 10 Stop

Ship information
Datetime: 2000/01/01 08:00:00
Interval(ms): 100 Play

Hotspot
Datetime: 2000/01/01 08:00:00
Interval(ms): 15 Play

Speed over ground
1 kn ~ Chg.

Figure 5: Top: Micro-level visualization of vessel movement. Bottom: Macro-level visualization of vessel movement.

α-strategies select agents from open list Θ_o. β-strategies select winning agents from winner list Θ_w. Choices of α-strategies and β-strategies include *worst*, *best*, *random* and *probabilistic* strategies. α-strategies and β-strategies select agent based on search criteria s_c. We have path quality $q(\pi_i, t_s)$ and path cost $c(\pi_i, t_s)$ of path π_i as the search criteria.

The *worst* strategy picks agent i such that $i = arg\min_{i \in \mathcal{N}}(q(\pi_i))$ when $s_c \equiv q(\pi_i)$ or $i = arg\max_{i \in \mathcal{N}}(c(\pi_i))$ when $s_c \equiv c(\pi_i)$. The *best* strategy picks agent i such that $i = arg\max_{i \in \mathcal{N}}(q(\pi_i))$ when $s_c \equiv q(\pi_i)$ or $i = arg\min_{i \in \mathcal{N}}(c(\pi_i))$ when $s_c \equiv c(\pi_i)$. The *random* strategy pick agent i randomly without any consideration for search criteria s_c. The *probabilistic* strategy picks agent i at a probability determined using $e^{-\frac{\alpha_p}{s_c}}$ where $0.0 < \alpha_p < 1.0$. In the experiments, $\alpha_p = 0.5$ is used.

The performance of VCM are illustrated using a heterogeneous scenario. The heterogeneity of scenario is implemented using vessel types. The types of vessel are characterized by two static attributes and one dynamic attribute. For agent i, the circular ship domain is the static attribute defined using physical radius r_i and safety parameter λ_i. The dynamic attribute is cruising SOG v_i. Agent i is assumed to move at constant cruising speed v_i along path π_i.

The experiments simulate the use of VCM in *live* mode. Thus, the initial set of paths $\Pi(\mathcal{N}, 0)$ is planned using unique O-D pairs of 12 heterogeneous agents. The full set of attributes of the heterogeneous agents are presented using Table 1. The unique O-D pairs leads to $\Pi(\mathcal{N}, 0)$ illustrated in Fig. 5.1. Also, for ease of comparison, the search process is performed for a fixed number of iterations rather than stopping when $\mathcal{Q}(\Pi(N), t_s - 1) \equiv \mathcal{Q}(\Pi(N), t_s)$. The experiments are performed for 5 runs. Experiment results from a run of the experiment are averaged using sliding window of size 2. After that, the experiment results are further averaged of 5 runs of the experiments.

Table 1: Attributes of heterogeneous agents.

Agent ID	Vessel Type	Physical Radius r_i	SOG v_i
1, 2	VLCC	2	1
9, 10	VLCC	2	2
3, 4	SuezMax	3	2
11, 12	SuezMax	3	3
5, 6	AfraMax	4	3
7, 8	ULCC	5	4

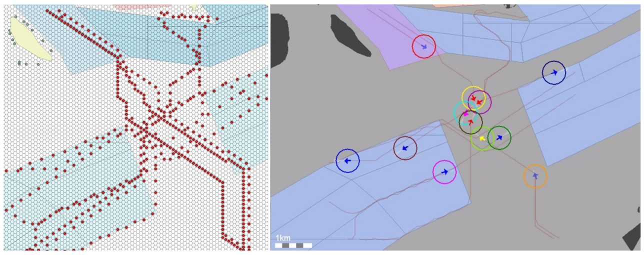


Figure 6: Illustration of uncoordinated paths $\Pi(\mathcal{N}, 0)$.

The ENC features used here are shown in Table 2. The ENC features defining the navigable areas are FAIRWY, TSSLPT, TSEZNE, TSSLPT:ORIENT, RECTRC and NAVLINE [5]. Areas not defined with any ENC features are referred to as the unregulated areas in Table 2. Vessels are not encouraged to be in these areas. So, a high cost value is given to discourage use of nodes from these areas in the paths.

Table 2: List of ENC features used in this work.

Acronym	ENC Feature	Cost
BERTHS	Place where a vessel is moored at a wharf	1
Navigable areas	Area of vessel navigate in	5
Unregulated areas	Areas where vessels are not preferred to be in	50
PRCARE	Area where vessels must navigate with caution	10
DMPGND	Dumping Ground	100
ANCHOR	Anchorage area	100

The experiment results are presented as the amount of improvement $\Delta\%$ of $\mathcal{Q}(\Pi(\mathcal{N}, t_s))$ from baseline $\mathcal{Q}(\Pi(\mathcal{N}, 0))$. Such an approach requires $\mathcal{Q}(\Pi(\mathcal{N}, 0))$ to be zeroed. At search iteration t_s, improvement $\Delta\%$ of $\mathcal{Q}(\Pi(\mathcal{N}, t_s))$ is calculated using

$$\Delta\%(\mathcal{Q}(\Pi(\mathcal{N}, t_s))) = 1.0 - \frac{1.0 - \mathcal{Q}(\Pi(\mathcal{N}, t_s))}{1.0 - \mathcal{Q}(\Pi(\mathcal{N}, 0))} \quad (6)$$

5.2 Experimental Results

Four sets of experiment results are presented here. Each set of experiment results compares the performance of VCM using different choices of key search parameters mentioned in Section 5.1. Such an empirical evaluation of VCM is necessary for understanding its underlying characteristics.

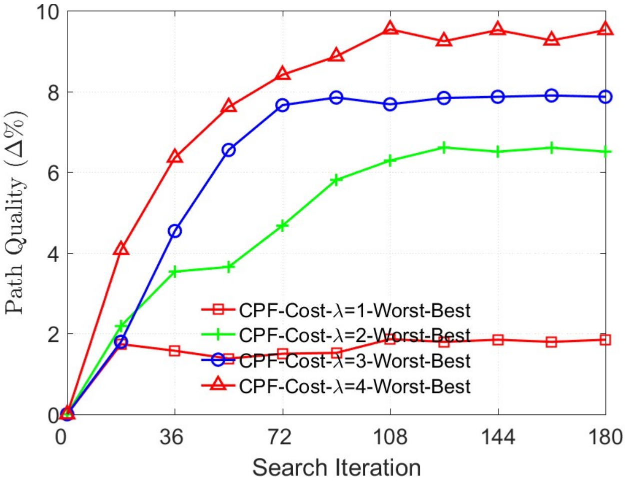


Figure 7: Comparison of experiment results generated using different safety parameter λ.

The plots of experiment results showing the performance of VCM using safety parameter $\lambda = \{1, 2, 3, 4\}$ are seen in Fig. 7. It is observed from Fig. 7 that larger λ gives better $\Delta\%$ of $\mathcal{Q}(\Pi(\mathcal{N}, t_s))$. This is expected because by looking further ahead, joint state space is larger and more distant vessels can be considered during path finding. Thus, paths are planned on more broadly updated $\mathcal{G}(t_s)$ whose cost of nodes can lead to *earlier* diversion compared to more narrowly updated $\mathcal{G}(t_s)$.

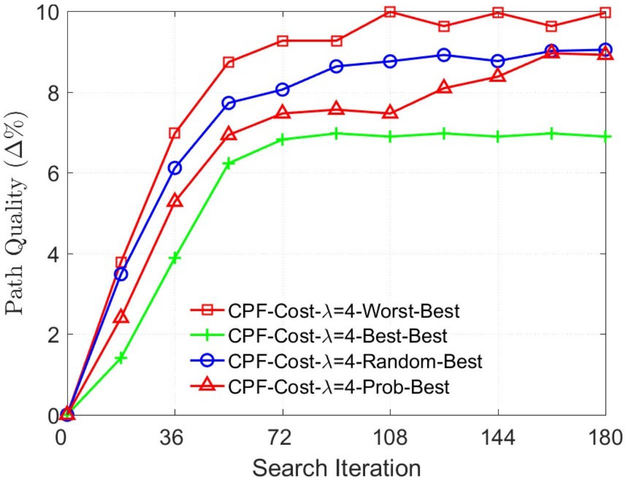


Figure 8: Comparison of experiment results generated using different α-strategies.

The plots of experiment results based on four α-strategies are seen in Fig. 8. The best performance is observed using *worst* α-strategy. In contrast, it is ineffective to pick agent using *best* α-strategy. This is because paths with worse path quality are expected to have larger $\Delta\%$ of $\mathcal{Q}(\Pi(\mathcal{N}, t_s))$ than paths with better path quality. In addition, *probabilistic* α-strategy is observed to be improving $\Delta\%$ of $\mathcal{Q}(\Pi(\mathcal{N}, t_s))$ rather slowly.

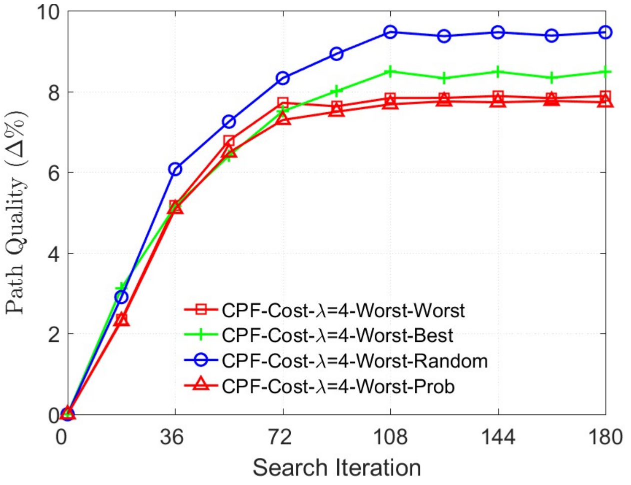


Figure 9: Comparison of experiment results generated using different β-strategies.

The plots of experiment results generated using four β-strategies are seen in Fig. 9. Comparing to Fig. 8, β-strategies appears to have lesser effect on $\Delta\%$ of $\mathcal{Q}(\Pi(\mathcal{N}, t_s))$. The *worst* and *probabilistic* strategies are observed having similar performance. The *best* β-strategy has just slightly better $\Delta\%$ of $\mathcal{Q}(\Pi(\mathcal{N}, t_s))$ than these two β-strategy. Rather surprisingly, the *random* β-strategy is observed having the best $\Delta\%$ of $\mathcal{Q}(\Pi(\mathcal{N}, t_s))$.

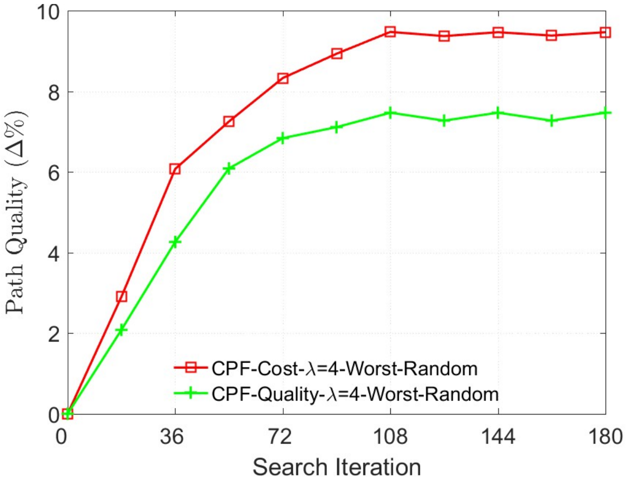


Figure 10: Comparison of experiment results generated using different search criteria s_c.

The plots of experiment results generated using the path cost $c(\pi_i, t_s)$ and the path quality $q(\pi_i, t_s)$ are seen in Fig. 10. Better performance is observed using $c(\pi_i)$ rather than $q(\pi_i)$ as the search criteria. This is due to the fact $c(\pi_i)$ is derived using the *physical radius* r_i and *safety parameter* λ_i while $q(\pi_i)$ is derived using just r_i. This means $c(\pi_i)$ has broader view of planning horizon. However, a trade-off for having a broader view of planning horizon is the increased cost of updating the cost of more nodes during path finding. Last but not least, Fig. 11 compares the paths and positions of the uncoordinated vessels and the vessels coordinated using $c(\pi_i)$ as the search criteria.

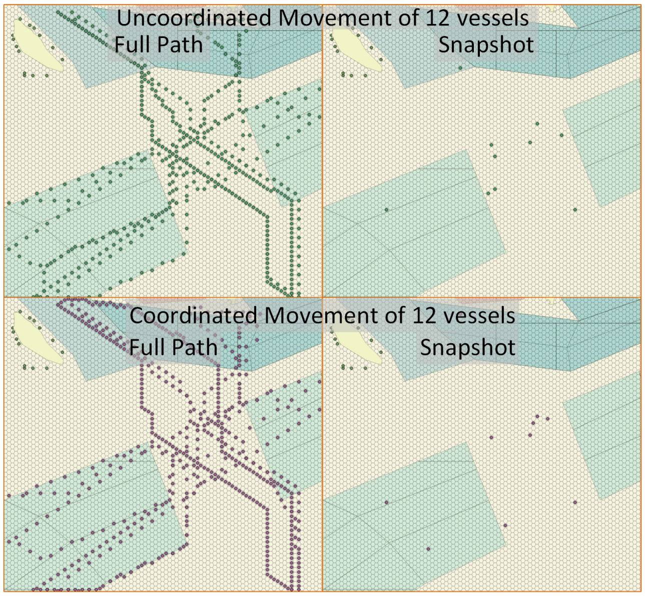

Uncoordinated Movement of 12 vessels
Full Path
Snapshot
Coordinated Movement of 12 vessels
Full Path
Snapshot

Figure 11: Left: Compares the full path of uncoordinated and coordinated movement of 12 vessels. Right: Compares the position of the uncoordinated vessels and the coordinated vessels at the same time point.

6. CONCLUSIONS

This paper presents our work on addressing the problem of coordinating the movement of vessel traffic formulated as a multi-agent path-finding (MAPF) problem with added complexity of maritime domain. This is because a multi-agent system in maritime domain is distributed and heterogeneous. The vessels have different ship domains, speed-over-ground (SOG) and movement characteristics. Unlike conventional MAPF, a vessel, represented as an agent, occupies multiple nodes in joint state space $\mathcal{G}$. In addition, the cost of nodes in $\mathcal{G}$ is based on time-dependent and time-independent features. Different agents can move at different SOGs along its path.

To address such a MAPF problem, this paper introduces an innovative application of search algorithm known as the Vessel Coordination Module (VCM). VCM has a search algorithm known as the Coordinated Path Finding (CPF) Algorithm. CPF algorithm generates coordinated path by checking for intersection of neighborhood of nodes on path of agents. It performs a neighborhood search at each solution point to identify the winning agents. During the neighborhood search, agents are selected for generating trial paths. Agents whose trial path improve the aggregated path quality are regarded as the winning candidates. After each neighborhood search, a winning agent is selected from the list of winning candidates. The joint state space is modified each time a trial path is found and when an agent is committed to a new path. New paths are found using joint state space updated using different sets of paths.

Experiments are conducted for evaluating the performance of VCM using different choices of the key search parameters comprising strategies for selecting agents during neighborhood search, strategies for selecting winning agents, safety parameter and search criteria. The experiment results identifies the set of key search parameters that gives coordinated paths with the best possible aggregated path quality. In addition, we have also illustrated the coordinated paths of vessels using a GIS visualization tool.

Our work can be continued in any of the following ways. Though VCM is shown to be effective for coordinating movement of vessels, there is still need to scale it up to handle larger number of vessels. We have also planned to use risk as a search criteria and have the reduction of overall risk as a performance metric. This is still an ongoing work because the maritime domain is far more sophisticated than presented here. Further details such as vessel movement models, more environmental and regulatory constraints can be included to improve fidelity of our work and can be used with higher level of confidence of its efficacy in the real world.

Acknowledgments

This research project is funded by National Research Foundation Singapore under its Corp Lab @ University scheme and Fujitsu Limited.

REFERENCES

[1] H. Andersson, K. Fagerholt, and K. Hobbesland. Integrated maritime fleet deployment and speed optimization: case study from RoRo Shipping. *Computers & Operations Research*, 55:233–240, 2015.

[2] M. Čáp, P. Novák, and A. Kleiner. Finding near-optimal solutions in multi-robot path planning. In *arXiv preparint arXiv:1410.5200*, pages 1–8. 2014.

[3] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[4] A. N. Ince and E. Topuz. Modelling and simulation for safe and efficient navigation in narrow waterways. *Journal of Navigation*, 57:53–71, 2004.

[5] International Hydrographic Bureau. S-57 IHO Object Catalogue. http://www.s-57.com, 2017.

[6] M. Jakob, O. Vaněk, and M. Pěchouček. Using agents to improve international maritime transport security. *IEEE Intelligent Systems*, 26(1):90–96, 2011.

[7] P. Janovský, M. Čap, and J. Vokřínek. Finding coordinated paths for multiple holonomic agents in 2-D polygonal environment. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, pages 1117–1123. 2014.

[8] Š. Kopřiva, D. Šišlák, and M. Pěchouček. Towards parallel real-time trajectory planning. *Advances on Practical Applications of Agents and Multi-Agent Systems*, pages 99–108, 2012.

[9] Maritime Port Authority of Singapore. STRAITREP. http://www.mpa.gov.sg/web/portal/home/port-of-singapore/port-operations/vessel-traffic-information-system-vtis/straitrep/description-of-straitrep, 2017.

[10] A. Napolitano, D. D. Crescenzo, and S. Gelli. A SelexSI solution to enable distributed decision making in vessels traffic management. In *Proceedings of the International Conference on System of Systems Engineering*, pages 203–208. 2012.

[11] L. Parrott, C. Chion, C. C. A. Martins, P. Lamontagne, S. Turgeon, J. A. Landry, B. Zhens, D. J. Marceau, R. Michaud, G. Cantin, N. M. enard, and S. Dionne. A decision support system to assist the sustainable management of navigation activities in the St. Lawrence River Estuary, Canada. *Environmental Modelling & Software*, 26:1403–1418, 2011.

[12] M. Pěchouček and V. Mařík. Industrial deployment of multi-agent technologies: review and selected case studies. *Journal of Autonomous Agent and Multiagent Systems*, 17:397–431, 2008.

[13] G. Praetorius. Control and resilience within the maritime traffic management domain. *Journal of Cognitive Engineering and Decision Making*, 8(4):303–317, December 2014.

[14] U. Simsir, M. F. Amasyali, M. Bal, U. B. Çelebi, and S. Ertugrul. Decision support system for collision avoidance of vessels. *Applied Soft Computing*, 25:369–378, 2014.

[15] D. Šišlák, P. Volf, and M. Pěchouček. Acceleted A^* Trajectory Planning: Grid-based Path Planning Comparison. In *Proceedings of the International Conference on Automated Planning* & *Scheduling*, pages 74–81. 2009.

[16] UK Hydrographic Office. *Passage Planning Guide - Malacca and Singapore Straits*. Witherby Publishing Group, 2015.

[17] I. Watanabe, H. Suzuki, T. Saitou, and S. Itoh. Macro-level and micro-level visualization of vessel movement. GIS Visualization Toolkit, 2016.