Agent-based Model of Maritime Traffic in Piracy-affected Waters

Ondřej Vaněk, Michal Jakob, Ondřej Hrstka, Michal Pechouček

Department of Computer Science and Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Karlovo Náměstí 13, Praha 127 00, Czech Republic

Abstract

Contemporary maritime piracy presents a significant threat to global shipping industry, with annual costs estimated at up to US$7bn. To counter the threat, policymakers, shipping operators and navy commanders need new data-driven decision-support tools that will allow them to plan and execute counter-piracy operations most effectively. So far, the provision of such tools has been limited. In cooperation with maritime domain stakeholders, we have therefore developed AGENTC, a data-driven agent-based simulation model of maritime traffic that explicitly models pirate activity and piracy countermeasures. Modeling the behavior and interactions of thousands of individually simulated vessels, the model is capable of capturing the complex dynamics of the maritime transportation system threatened by maritime piracy and allows assessing the potential of a range of piracy countermeasures. We demonstrate the what-if analysis capabilities of the model on a real-world case study of designing a new transit corridor system in the Indian Ocean. The simulation results reveal that the positive past experience with the transit corridor in the narrow Gulf of Aden does not directly translate to the vast and open waters of the Indian Ocean and that additional factors have to be considered when designing corridor systems. The agent-based simulation development and calibration process used for building the presented model is general and can be used for developing simulation models of other maritime transportation phenomena.

Keywords: agent-based simulation, computational modeling, maritime transportation, maritime piracy, policy assessment
1. Introduction

Global maritime shipping lanes are a critical part of the world’s transportation infrastructure. 90% of internationally traded goods are transported by sea at at least one point in their journey (Earnest and Yetiv, 2009). In the past years, the global maritime transportation system has come under a serious threat from maritime piracy. As a major security and economic threat costing the global economy up to estimated US$7bn (Bowden et al., 2011), contemporary maritime piracy has solicited a concerted international response which has, finally, led to the reduction of the success rate of pirate attacks. The number of pirate attacks, average amount of ransom paid and the number of seafarers held in captivity, however, remain high. In 2011 and for Somali-based piracy alone, there were 181 attacks reported for Somalia, 28 vessels hijacked and 1118 seafarers were taken hostage\(^1\). Containing piracy also required and continues to require extensive deployment of naval forces which is unsustainable in a long term.

From the many levels on which solutions of the problem are sought, we focus on the operational management of the situation at sea, as this is the arena where progress can be made in the short term, before long-term sustainable solutions can be developed onshore. To date, military, governmental and industry stakeholders have proposed several types of piracy countermeasures to increase the security of maritime transit, including recommended transit corridors, group transit, escorted convoy schemes, coordinated patrol deployments and on-board security teams. When properly designed and implemented, such measures can significantly improve maritime transportation security with reasonable additional cost. However, due to complex spatial and temporal dependencies between individual countermeasures and external factors, discovering effective, synergistic combinations of piracy countermeasures presents a major challenge.

To address this challenge, we have built AGENTC, a data-driven agent-based simulation model of maritime activity in piracy-affected waters. The model aims at helping decision makers reduce uncertainty about the effects of their operational control and regulatory interventions. The model incorporates a wide range of real-world data and, to our best knowledge, is the first computational model that simulates deep sea shipping down to the level of individual vessels. This is crucial for accurately capturing emergent, collective effects arising from the context-dependent interactions of merchant, pirate and navy vessels.

Due to the lack of prior work on the topic, the development of the model prompted the development of a novel methodology for agent-based maritime transportation modeling. Some parts of the methodology could be borrowed from more mature transportation modeling fields; other parts had to be developed from scratch. In addition to the AGENTC model, the developed method-\(^1\)\footnote{Source: International Maritime Bureau (IMB) Piracy Reporting Centre (website: http://www.icc-ccs.org/piracy-reporting-centre).}
ology, presented alongside the model itself, is therefore the second major contribution of the paper.

The rest of the paper is organized as follows. After reviewing related work, we describe the AGENTC simulation model, detailing how the maritime environment, vessel behaviors and vessel interactions are modeled. We then briefly comment on the implementation aspects of the simulation model and devote significant space to discussing calibration of the model. Finally, we show how the developed model was employed to help answering specific operations research questions concerning the design of maritime transit corridor systems.

2. Related Work

The use of agent-based or simulation-based models to support policy design and operational management has a very long-standing tradition in the transportation field. The vast majority of the work, however, focuses on ground transportation (e.g. Hidas, 2002; Waraich et al., 2013) and, to a lesser extent, on air transportation (e.g. Tang et al., 2012).

In the maritime domain, applications of simulation models are surprisingly scarce, as analyzed, e.g., by Davidsson et al. (2005). Existing work either focuses on traffic in ports and national, coastal waters (Hasegawa et al., 2004) or uses high-level equation-based models (Bourdon et al., 2007) unfit for capturing individual-level behavior and inter-vessel interactions essential for modeling maritime piracy. Furthermore, none of the above models is concerned with the security of maritime shipping lanes. Advanced computational methods have been applied in the maritime domain to optimize ship routing (e.g. Norstad et al., 2011; Øvstebø et al., 2011), albeit not taking the security aspect into account. In both cases, the authors use mathematical programming rather than simulation to solve the problem.

As far as the security angle on transportation systems is concerned, existing simulations focus on modeling activities in and around terminals rather than within transportation networks themselves. This is true both for airport security (Chawdhry, 2009) and port security (Koch, 2007). Port security has also been listed as an important area for the application of operations research methods (Crainic et al., 2009), of which simulations are an important representative. The spatial, network aspect of transportation security has been touched upon in the work on modeling critical infrastructures (Barton and Stamber, 2000), however, the emphasis there is mostly on other than transportation types of infrastructures. The problem of securing transportation infrastructures and logistical networks has only been studied in the military context (Ghanmi et al., 2011).

Focusing on the very phenomenon of maritime piracy, existing work is concentrated primarily in the fields of security studies, international relations and global policy (Omoha, 2010). Only recently, initial attempts at applying computational modeling and optimization to maritime piracy have emerged but focus exclusively on military aspects of the problem: Bruzzone et al. (2011) model piracy around the Gulf of Aden using the discrete-event simulator PANOEPA.
The authors focus on evaluating the efficiency and effectiveness of different Command and Control models; only main actors in the Gulf of Aden are considered and the simulation is not scaled to the Indian Ocean where the merchant traffic model is significantly more complicated.

Tsilis (2011) employs the MANA agent-based modeling framework (Lauren and Stephen, 2002) to identify key factors affecting the escort of vulnerable merchant vessels through the Gulf of Aden. The escorting scenario is modeled on a tactical level, focusing on positioning of individual ships and protection of one group of merchant vessels; this is different from our model which adopts a whole-system perspective and considers the security of maritime transportation system as a whole. The MANA framework is also used by Decraene et al. (2010) to analyze requirements on non-lethal deterents for defending large merchant vessels against pirate attacks; again, the focus is on the tactical level of modeling a single encounter in detail, rather than the system as a whole.

Slootmaker (2011) describes Next-generation Piracy Performance Surface (PPSN) model which employs meteorological forecasts, intelligence reports and historical pirate incidents to predict areas conducive to pirate activity around the Horn of Africa. Hansen et al. (2011) further improve the PPSN model by refining the environment model and adding a probabilistic behavioral pirate model, resulting into the Pirate Attack Risk Surface (PARS) model. Both PPSN and PARS models are numerical with only a minor simulation component and are limited to short-term forecasts (several days). They do not directly model real-world behavior and interactions of individual vessels; consequently, their applicability for what-if type of analysis is limited.

Finally, piracy patterns and the effect of countermeasures were also studied using statistical data analysis and data mining (Bowden et al., 2011). The usability of such results for policy design and optimization is limited because the insights gained concern the behavior of the maritime system under current circumstances and are difficult to extrapolate to hypothetical future scenarios.

3. Model Description

The AgentC model represents the movement and other activities of selected categories of vessels in piracy-affected waters of the Indian Ocean (we focus on piracy with origins in Somalia, affecting Gulf of Aden, Arabian Sea and West Indian Ocean). A visual overview of the model and its constituent entities is given in Figure 1.

The description of the model proceeds as follows: after explaining the agent-based modeling methodology, we describe the model of the maritime environment and models of each vessel class in detail. Given the importance of vessel interactions, we separately describe the model of the pirate attack and of selected piracy countermeasures.

3.1. Agent-based Modeling Methodology

As already stated, we employ individual-centric modeling approach, in which the behavior of the modeled system is represented at the micro-level of indi-
individual vessels. Vessels are modeled as autonomous agents (Russell et al., 2010) capable of moving freely within the navigation boundaries of ocean waters while interacting with the maritime environment, other vessel agents and other actors (such as shipping operators or traffic coordinators).

Based on the literature and discussions with domain experts, we identified merchant vessels, pirate vessels and navy vessels as main vessel classes which are therefore explicitly represented in our model as vessel agents. For most of the time, each vessel agent pursues its individual goals, however, there are situations where multiple vessel agents interact—such interactions are either non-cooperative (such as pirate attacks or navy warship counter-pirate interventions) or cooperative (such as merchant vessel agents’ requests for help to navy vessel agents). Vessel agent interactions play a critical role in the dynamics of maritime piracy and make the agent-based, micro-simulation approach vital for accurately modeling the effect of piracy on maritime transportation, primarily because it allows capturing the phenomena naturally and it provides the detail of analysis not attainable with macro-level equation-based methods (Van Dyke Parunak et al., 1998).

In line with the agent-based modeling approach, the model for each class of vessels consists of an individual vessel behavior model and a vessel population model. The vessel behavior model represents the executable behavior of an individual vessel agent; such behavior can depend on vessel parameters assigned to each individual vessel. The vessel population model specifies how many vessels of each class are generated and how the values of vessel parameters are assigned.

The choice of vessel parameters is based on relevant literature (Bruzzone et al., 2011; Tsilis, 2011; Decraene et al., 2010) and was discussed with domain experts; their influence (i.e., importance) was also explored using sensitivity analysis (described in Section 5.2). For each vessel class, we list the parameters in a table together with the intervals of parameter values considered in the simulation. For each simulation instance, depending on the scenario executed, we either sample the value of a parameter uniformly from its respective interval or set the parameter value to a constant from the interval.

Note on Naming

In the military domain, merchant vessels are denoted as MV, pirate vessels—recently mainly operating as a group of one mothership and multiple accompanying speedboats—are denoted as pirate attack group (PAG) and navy vessels as coalition forces (CF). We denote the vessel classes as M, P and N respectively. Furthermore, although we use the term vessel and vessel agent rather interchangeably, we use the latter if we want to stress the behavioral aspect of vessel description. With some simplification, vessel agent can be viewed as the shipmaster controlling a respective vessel.

3.2. Maritime Environment Model

The environment model represents physical maritime environment in which the vessels operate. It consists of two principal components:
Figure 1: Key actors, activities and environmental features represented in the AgentC model of piracy affected waters.

- **geography, bathymetry**—represent the geography of the maritime environment in terms of a set of spherical obstacle polygons $O$ representing land masses, shallow waters and other obstructions that limit navigability. This component also contains locations of ports and anchorages used in merchant shipping and pirate activities.

- **weather**—represents the environmental conditions affecting the behavior of modeled vessels, specifically, wave height, wind speed and currents. Wave height plays an important role in pirate’s decision making; currents and wind slightly alter routes of small vessels (e.g., pirate boats).

### 3.3. Merchant Shipping Model

Merchant vessels are large ocean-going vessels carrying cargo over long distances between world’s major ports. Merchant vessels are the primary targets of pirate attacks. In order to be useful for what-if analysis of different counterpiracy measures and policies, the merchant traffic model has to produce realistic traffic patterns even for situations which diverge from the current status quo in terms of pirate operations and the configurations of piracy countermeasures deployed. The merchant traffic model therefore cannot solely mimic current real-world merchant vessel routes but it has to be capable of generating realistic shipping traffic from more fundamental principles. We therefore adopt the approach of separating the modeling of transportation demand from traffic routing. Demand is considered given and fixed while the routes are generated dynamically, based on the assumption that merchant vessel agents maximize their utility and take the most advantageous route possible. Such an approach is widely used in ground transportation modeling, its application to global maritime shipping, however, is novel.
Merchant Shipping Origin–Destination Matrix

The demand for merchant transportation is represented in terms of an origin–destination matrix (O-D matrix) which specifies the volume of merchant traffic between world’s major ports. The O-D matrix is used to generate origins and destinations for individual merchant vessel voyages and the voyage planning module is then used to find optimum routes connecting voyage endpoints.

Unfortunately, in contrast to ground traffic modeling, no data explicitly and completely capturing the merchant shipping O-D matrix is available; we were therefore forced to estimate the matrix from several partial sources. We have extracted the most important ports in and near the observed area from CI-online database and Ports&Ships portal. We then estimated the O-D matrix by fitting generated traffic to known real-world traffic densities (see Section 5.3).

Voyage Planning

A fundamental part of the merchant vessel operation is voyage planning. Voyage planning is primarily used in the model of merchant vessels to generate realistic merchant traffic from the merchant shipping O-D matrix; however, it is also used in the operation of the other two vessel classes.

We model voyage planning as an optimization problem of finding an optimal route $C$ between an origin and a destination point on a sphere, given vessel-specific route optimality criterion, a set of constraints imposed by geographical boundaries and physical properties of the vessel, and a spatial piracy risk function (the latter is only used in voyage planning for merchant vessels).

Mathematically, we formalize the optimal route selection problem as a bi-objective optimization problem

$$
\min \quad L(C), R(C) \\
\text{s.t.} \quad C \in \mathcal{C}
$$

(1)

where $L(C)$ is the length of route $C$ on the sphere, $\mathcal{C}$ is the set of all valid routes, i.e., routes going from the origin to the destination respecting geographical boundaries, and $R(C)$ is the risk of a pirate attack along route $C$, computed as

$$
R(C) = \int_C r ds
$$

(2)

Here,

$$
r(x, y, t) : \mathbb{R}^3 \to \mathbb{R}_0^+
$$

(3)

is termed the piracy risk function and captures the expected density of pirate attacks for a given time and location, defined by latitude $x$ and longitude $y$. The piracy risk function can be constructed based on past pirate incident reports$^3$.

$^2$http://www.ci-online.co.uk, http://ports.co.za/

$^3$as provided e.g. by the International Chamber of Commerce Piracy Reporting Centre at http://www.icc-ccs.org/
or another risk model in the form of a spatio-temporal or spatial function can be used, e.g., the NATO Shipping Centre\(^4\) pirate activity map.

To solve the optimization problem (1), we transform the length and risk criteria into a scalar criterion function using the aggregation method (Hwang et al., 1979) with a single weight:

\[
\min \ (1 - \alpha) \mathcal{L}(C) + \alpha \mathcal{R}(C) 
\]  

(4)

The weight \(\alpha\) is termed the risk aversion coefficient and it can be set individually for each merchant vessel agent based, e.g., on the level of on-board security, vessel cruising speed or the value of its cargo.

In order to leverage efficient path-finding algorithms, we discretize the ocean surface space and represent it as a graph. First, we define the piracy risk area as a spherical polygon \(O_R\) which delineates the space in which piracy attacks occur and where we explicitly consider piracy risk in the routing process. For discretization, we divide the piracy risk area \(O_R\) into a rectangular latitude-longitude grid \(G_r\) (Sahr et al., 2003) with predefined cell widths. We create a graph \(G_1(V_1, E_1)\) from the grid \(G_r\): the set of vertices \(V_1\) comprises all vertices from the grid and the set of edges \(E_1\) is the set of all geodesics\(^5\) between any two neighboring vertices (considering 16-connected cells (Boult et al., 1993); see Figure 2a).

Second, we construct a spherical visibility graph \(G_2(V_2, E_2)\). The vertices \(V_2\) of the visibility graph comprise all vertices of all spherical polygons \(O \cup O_R\) (\(O\) is the set of obstacle polygons, see Section 3.2). The set of edges \(E_2\) of the visibility graph is the union of the set of all geodesics between all vertices \(V_2\) which do not intersect any polygon from \(O \cup O_R\) (Figure 2b) and the set of entry and exit edges, where the entry and exit edges are all such edges connecting the voyage’s origin and destination point to the visibility graph \(G_2\) that do not intersect any polygon from \(O \cup O_R\) and any edge from \(E_1\). We set the risk value on edges outside the piracy risk area \(O_R\) to 0; for edges inside the piracy risk area \(O_R\), we set the risk value to the integral of the piracy risk function (3).

We compute the optimum vessel route in graph \(G(V_1 \cup V_2, E_1 \cup E_2)\) with respect to criterion (4) using the \(A^*\) algorithm (Russell et al., 2010) with orthodromic distance\(^6\) heuristics and with the cost function equal to the criterion function. For illustration, vessel routes generated by the route planner between all pairs of ports considered in the AGENTC model are shown in Figure 2c. Risk aversion coefficient \(\alpha = 0.5\) and a spatial risk model obtained from the NATO Shipping Centre were used.

\(\text{Merchant Vessel Population Model}\)

The merchant vessel population model instantiates a population of merchant ships of size \(#M\) with a realistic distribution of key vessel attributes, i.e., speed

\(^4\)NATO Shipping Centre website: http://www.shipping.nato.int
\(^5\)The shortest path between two points on a surface of a sphere.
\(^6\)The shortest distance between any two points on the surface of a sphere.
Figure 2: (a) rectangular grid representing the piracy risk area, (b) cell-grid connected to the visibility graph, (c) PARS model for October 1st, 2011, provided by NATO Shipping centre—green, yellow, red colors correspond to low, medium and high risk of attack respectively; plans are generated for each O-D matrix entry with risk aversion coefficient $\alpha = 0.5$.

and size. It then samples the merchant shipping O-D matrix in order to assign each vessel a port of origin and a destination port to reach. All merchant vessel attributes specified by the population model are listed in Table 1. 2000 merchant vessels with speed and size distribution taken from a data-set of 2700 real-world vessel samples\(^7\) were used in the simulation to replicate real-world shipping density in the Indian Ocean in Year 2011.

**Merchant Vessel Agent Behavior Model**

The behavior of a simulated merchant vessel is straightforward. Given a pair of origin-destination ports at the beginning of the simulation, the merchant vessel agent invokes the route planner to plan vessel’s voyage, taking into account corridors and group transit schemes along its route. The merchant vessel then sets on cruising along the route. After the destination port is reached, a new port is sampled from the O-D matrix and a new route is planned. This basic behavior is interrupted if the vessel is attacked by a pirate vessel, in which case the merchant vessel agent reports attack to nearby merchant vessel agents, notifies the closest navy vessel agent and employs self-defense measures (see Section 3.6). Activities and transitions of the merchant vessel agent behavior are depicted in Figure 3.

**3.4. Navy Operations Model**

Navy vessels represent military vessels operating in piracy-affected waters and capable of using force to deter and disrupt pirate activities. AGENTC focuses on modeling the part of Navy vessel operation consisting in providing assistance to vessels subject to pirate attacks. It does not model active search

---

\(^7\)The data-set is a subset of the Vesseltracker ([http://www.vesseltracker.com](http://www.vesseltracker.com)) database.
Table 1: Merchant vessel parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination</td>
<td>port id</td>
<td>Destination port of vessel voyage</td>
</tr>
<tr>
<td>Docking time</td>
<td>[0, 3] days</td>
<td>Docking time of a merchant vessel</td>
</tr>
<tr>
<td>Cruising speed</td>
<td>[10, 20] kn</td>
<td>Vessel travel speed unless participating in a group transit</td>
</tr>
<tr>
<td>Ship size</td>
<td>[30, 250] m</td>
<td>Size of the ship</td>
</tr>
<tr>
<td>Alertness</td>
<td>[0, 60] hr⁻¹</td>
<td>Frequency of checking for an approaching pirate</td>
</tr>
<tr>
<td>Risk aversion</td>
<td>[0, 1] -</td>
<td>Risk aversion coefficient used in voyage planning</td>
</tr>
</tbody>
</table>

Figure 3: Merchant vessel agent behavior model. The entry point is the Dock/Wait state. After docking in a port, the merchant vessel plans a route and cruises to its destination. If a pirate is spotted, a request for help is sent. In case the vessel is hijacked, the hijacked state is terminal and the merchant vessels is under the control of the pirate.
Table 2: Navy vessel parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helicopter</td>
<td>Y/N</td>
<td>Presence of helicopter on board the navy vessel</td>
</tr>
<tr>
<td>Patrolling</td>
<td>GPS Coords.</td>
<td>Area at which the navy vessel is located and from where it can respond</td>
</tr>
<tr>
<td></td>
<td></td>
<td>to nearby pirate attack</td>
</tr>
<tr>
<td>Action radius</td>
<td>[100, 200] nm</td>
<td>Distance on which the navy vessel reacts to distress calls</td>
</tr>
<tr>
<td>Response speed</td>
<td>[20, 30] kn</td>
<td>Speed at which the vessel sails to intercept pirate attack</td>
</tr>
<tr>
<td>Helicopter Speed</td>
<td>[140, 170] kn</td>
<td>Speed of the on-board helicopter</td>
</tr>
</tbody>
</table>

and area patrolling operations, although such extensions can easily be incorporated in the model assuming data about such operations (which are typically classified) are obtained.

**Navy Vessel Population Model**

Navy vessel population model instantiates #N navy vessel agents with specified deployment locations. The deployment locations can be specified manually by a human expert or obtained as a result of an optimization process (see below). All navy vessel parameters are listed in Table 2.

**Navy Vessel Behavior Model**

The basic behavior of a navy vessel comprises staying in its deployment location waiting for possible distress calls from nearby merchant vessels threatened by pirates. If a distress call is received, the navy vessel agent responds by dispatching a helicopter (if available) and by moving at its cruising speed to the attacked merchant vessel, trying to intercept the attack. Once the response has been completed, the navy vessel returns to its original deployment position. More details about navy vessel involvement in pirate attacks are given section 3.6. The behavior model of the navy vessel agent is depicted in Figure 4.

**Navy Vessel Location Assignment Model**

Similarly to the merchant traffic model, the navy operations model cannot solely replicate existing real-world deployment locations, but it needs to be able to take into account hypothetical merchant traffic flows in diverse evaluated what-if scenarios. We have therefore developed a navy vessel location assignment algorithm which takes the density of merchant traffic as the input and produces navy vessel deployment locations. The choice of locations aims to maximize the proportion of merchant traffic that lies within the action radius of deployed navy vessels.

We formalize the problem of optimal asset allocation as an optimization problem. We consider the merchant traffic density function $m(x, y) : \mathbb{R}^2 \to \mathbb{R}_0^+$.
Figure 4: Navy vessel agent behavior model. The entry point is the Patrolling state. The navy vessel reacts on a request for help by deploying a helicopter (if available) and cruises towards the merchant vessel. If the navy vessel or the helicopter arrives before the merchant vessel is hijacked, the pirate is disarmed and the vessel (and helicopter) returns to its assigned location. The intervention terminates unsuccessfully if the pirate successfully completes the hijacks of the merchant vessel.

specifying the average density of merchant vessels at a location specified by latitude $x$ and longitude $y$.

We assume that each navy vessel can protect merchant traffic that lies within the vessel’s action radius and that the probability that the navy vessel will prevent the attack decreases exponentially with the distance from the navy vessel. Specifically, we model the probability of attack prevention as

$$p(x, y | \mu_x, \mu_y) = e^{\exp\left(-\frac{(x - \mu_x)^2 + (y - \mu_y)^2}{2\sigma^2}\right)}$$

(5)

where parameters $\mu_x, \mu_y$ represent the lat/lon location of the navy vessel and we set $\sigma = \frac{1}{3}AR$ to approximate that the probability of successful attack prevention is very high near the navy vessel, declines sharply as the distance from the navy vessel increases and is almost zero once the attack happens outside the vessel’s action radius $AR$.

The navy vessel location assignment optimization problem is then formulated as

$$\arg \min_{\mu^1_x ... \mu^N_x, \mu^1_y ... \mu^N_y \in \mathbb{R}} \int \int_A m(x, y) \prod_{i \in 1...N} \left(1 - p(x, y | \mu^i_x, \mu^i_y)\right) dxdy$$

(6)

i.e., we search for the positions $\mu^i_x, \mu^i_y$ of all navy vessels such that the expected number of successfully attacked vessels is minimized.

Similarly to voyage planning, for practical reasons we solve the allocation problem in a discrete formulation. We introduce the merchant traffic density map $m_G : G_m \to \mathbb{R}_0^+$ where $G_m$ is a rectangular latitude-longitude grid covering the navigable ocean waters. For any cell $c \in G_m$, $m_G(c)$ specifies the number of merchant vessels passing through the cell for a specified time interval (e.g., a month or a year). Density maps for each month are provided by, e.g., AMVER8 or they can be generated for any time span from the AgentC model.

---

We define functions $\chi_x, \chi_y : G_m \to \mathbb{R}$ specifying the latitude $\chi_x(c)$ and the longitude $\chi_y(c)$ of the central point of cell $c \in G_m$.

The allocation algorithm searches for the set of cells that maximize the protection of merchant traffic. Specifically, we use an iterative greedy algorithm to determine navy vessel locations—in the $i$-th iteration, the $i$-th navy vessel is placed to the center of a cell $c^i$ such that the following expression

$$c^i = \arg \min_{c^* \in G_m} \sum_{c \in G_m} m_G(c) \cdot p(\chi_x(c), \chi_y(c) | \chi_x(c^*), \chi_y(c^*)) \cdot \prod_{j=1}^{i-1} p(\chi_x(c), \chi_y(c) | \chi_x(c^j), \chi_y(c^j)) \quad (7)$$

is minimized. The expression $\prod_{j=1}^{i-1} p(\chi_x(c), \chi_y(c) | \chi_x(c^j), \chi_y(c^j))$ captures the protection provided by navy vessels placed in preceding iterations.

### 3.5. Pirate Activity Model

Pirate vessels range from small skiffs up to large motherships acting as a floating base from which speedboats are launched to attack. We model piracy at the level of individual pirate attack groups which are represented by a single pirate vessel agent having its home anchorage and operating in and around main shipping lanes, where it attempts to attack, board and hijack passing merchant vessels.

#### Pirate Population Model

The pirate population model is currently simple and is only used to generate $\#P$ pirate agents and to assign them their *home-anchorage* parameter. The assignment is based on reported estimates of the number of pirate attack groups operating from each known pirate anchorage. All parameters of the pirate are listed in Table 3.

#### Pirate Vessel Agent Behavior Model

The pirate vessel agent behavioral cycle consists of three stages:

1. **Cruising**—the pirate vessel moves directly to its selected target area and looks for a suitable merchant vessel to attack. If the pirate vessel agent spots a navy vessel, it steers away temporarily. When the pirate vessel reaches the target area, it moves at a low speed and changes its course randomly from time to time.

2. **Attack**—If a suitable merchant vessel is spotted, pursuit starts (described in detail in Section 3.6).

3. **Recuperation**—After a successful attack or when running out of supplies (*endurance* parameter), the pirate agent navigates back to its home anchorage. After an unsuccessful attack, the pirate recovers (*cool-down time* parameter) and looks for a merchant vessel again.
### Table 3: Pirate vessel parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home anchorage</td>
<td>base id</td>
<td>Base from which the pirate vessel embarks and to which it returns</td>
</tr>
<tr>
<td>Cruising speed</td>
<td>[8, 14] kn</td>
<td>Normal speed when traveling long distance between the base and a target location</td>
</tr>
<tr>
<td>Pursuit speed</td>
<td>[25,30] kn</td>
<td>Speed during the attack on a merchant vessel</td>
</tr>
<tr>
<td>Endurance</td>
<td>[7, 21] days</td>
<td>The number of days the pirate vessel can stay at sea</td>
</tr>
<tr>
<td>Visibility radius</td>
<td>[5, 12] nm</td>
<td>Maximum distance of a merchant vessel which the pirate can spot</td>
</tr>
<tr>
<td>Attack time</td>
<td>30 min</td>
<td>Duration of attack attempt</td>
</tr>
<tr>
<td>Cool-down time</td>
<td>[1, 4] hr</td>
<td>Time needed for recovery after an unsuccessful attack</td>
</tr>
<tr>
<td>Navy knowledge</td>
<td>[0, 1]</td>
<td>Probability of knowledge about navy vessel position</td>
</tr>
<tr>
<td>Hijack prob. $\rho_u$</td>
<td>[0, 1]</td>
<td>Probability of successful hijack of a merchant vessel cruising at 10 nm unaware of the pirate attack</td>
</tr>
<tr>
<td>Hijack prob. $\rho_a$</td>
<td>[0, 1]</td>
<td>Probability of successful hijack of a merchant vessel cruising at 10 nm aware of the pirate attack</td>
</tr>
</tbody>
</table>

Activity diagram of pirate vessel agent behavior is depicted in Figure 5. Note that we do not model the economic aspect of piracy, such as ransom negotiation and other processes taking places after a hijacked vessel is brought to shore.

**Target Attack Area Selection Mechanism**

A key part of pirate vessel decision-making is choosing its target area where it will look for a merchant vessel to attack. Again, in order to have the pirate activity reflect the simulated scenario, target area cannot be predefined based on attack locations currently observed in the real world but needs to be determined dynamically from more fundamental principles.

**AGENTC** therefore implements an algorithm that determines attack locations dynamically, from the assumption of pirate’s rationality and partial knowledge of both merchant vessels and naval patrols: pirate’s target area is thus selected based on the following inputs: weather conditions, the merchant traffic density map in a form of a grid and a (partial) knowledge about navy vessel positions.

The algorithm for target area selection is described in Algorithm 1. In the first step, a subset of regions with acceptable weather conditions for a given date is selected\(^9\). In the second step, navy vessels positions—if known—are

---

\(^9\)Based on the correlation of attack frequencies and weather conditions in 2011, we have
Figure 5: Pirate vessel agent behavior model. The entry point is the Select target area state. After the pirate reaches the target area previously selected, it looks for a suitable merchant vessel to approach and hijack. The hijack attempt can be unsuccessful, interrupted by a navy vessel—in which case the pirate is disarmed and sails home—or successful, in which case he sails with the merchant vessel to its home base.

Algorithm 1 Target Area Selection Algorithm

1: regions ← selectRegions(date)
2: densityMap ← merge(NavyPos, MerchantMap)
3: cList ← getCells(densityMap, regions)
4: totalValue ← sumValues(cList)
5: for cell ∈ cList do
   cell.prob ← cell.value/totalValue
end for
6: cell ← sample(cList)
7: return cell

combined with the merchant traffic density map (this procedure is equal to the navy vessel location assignment algorithm with positions of navy vessels being already known) and only cells from the merchant traffic density map within the regions with the acceptable weather are considered (cList on the 3rd line of the algorithm).

Each cell from cList has a probability assigned which is proportional to the value of the cell taken from the merchant traffic density map, i.e., we divide the value of each cell by the sum of values of all cells in cList (line 5-6). Finally, a simple random sampling mechanism returns a cell from cList, i.e., each cell is chosen with a probability proportional to the value of the cell in the risk map.

The above location selection mechanism approximates the combination of two fairly complex driving forces behind the pirate’s decision-making process: (1) the pirate wants to maximize its expected reward and prefers high density

estimated the acceptable wave height for piracy operations to be up to 1.25 m.
cells to low density cells; (2) the pirate is subject to game-theoretic interaction in the sense that he does not want to be predictable (Tambe, 2011). These two counter-going objectives of pirate are captured by the weighted randomization over the merchant traffic density map with consideration of regions with unsuitable weather and known positions of naval vessels.

3.6. Pirate Attack Model

Pirate attack is a complex interaction between all three classes of vessels; we therefore provide its standalone description that complements the description of the attack from the perspective of individual vessel agent behaviors. Parameters directly influencing the course and the outcome of the attack are depicted in Table 4 and are a subset of parameters of individual vessel classes, except the M awareness binary parameter which is well-defined only during the attack phase. All interactions taking place during a pirate attack are depicted in Figure 6. The attack consists of three phases:

1. pre-attack/approach—this phase begins after a merchant vessel is spotted by a pirate vessel agent; the pirate vessel agent starts a pursuit at its pursuit speed. The merchant vessel agent checks for a, approaching pirate vessel several times per hour (parameterized by the alertness parameter capturing the probability of spotting an approaching pirate). If a pirate vessel is spotted during its approach, the awareness parameter is set to true, meaning that the merchant vessel is not taken by surprise by the attacking pirate and can deploy self-defensive countermeasures. Furthermore, upon spotting the attack, the attacked merchant vessel agent broadcasts a distress call and notifies nearby merchant vessel agents about the danger (their awareness is then set to true). If there is an idle navy vessel within the navy vessel action radius, the navy vessel responds by moving towards the attack. If the navy vessel carries an on-board helicopter, it dispatches the helicopter to prevent the pirate from hijacking the merchant vessel.

2. attack—the pirate vessel agent attempts (repeatedly, for a time period, specified by the attack-time parameter) to board the merchant vessel and seize control. The probability of success depends on the speed $s$ of the merchant vessel, its alertness $a$ and subsequently on its awareness. The average probability of a merchant vessel being hijacked without any navy vessels present in the model can be computed as $p_h = a \cdot p_a(s) + (1 - a) \cdot p_u(s)$, where $p_a(s)$ and $p_u(s)$ are the probabilities of hijacking an aware and unaware merchant vessel, respectively, traveling at speed $s$. The hijacking probabilities are linear functions (with threshold) of merchant vessel speed: $p_a(s) = \max\{0, (2 - s/\nu)p_a\}, p_u(s) = \max\{0, (2 - s/\nu)p_u\}$, where $p_a$ and $p_u$ are base probabilities specifying the probability of hijacking a merchant vessel cruising at $\nu \geq 10$ kn (minimum cruising speed of a merchant vessels in our model is 10 kn).

3. post-attack—if the attack is successful, the hijacked vessel is taken to the pirate’s home anchorage; if the attack is aborted by the pirate (after
Table 4: Parameters affecting the outcome of a pirate attack.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M Cruising speed</td>
<td>[10, 20] kn</td>
<td>Cruising speed of the merchant vessel</td>
</tr>
<tr>
<td>M Alertness</td>
<td>[0, 60] hr⁻¹</td>
<td>How often the merchant vessel checks for pirate presence in its vicinity</td>
</tr>
<tr>
<td>M Awareness</td>
<td>Y/N</td>
<td>Merchant vessel’s knowledge about an approaching pirate (element of surprise)</td>
</tr>
<tr>
<td>P Visibility radius</td>
<td>[5, 12] nm</td>
<td>Maximum distance at which a merchant vessel can be spotted</td>
</tr>
<tr>
<td>P Pursuit speed</td>
<td>[25,30] kn</td>
<td>Cruising speed of the pirate</td>
</tr>
<tr>
<td>P Attack time</td>
<td>30 min</td>
<td>Maximum time for which the pirate attacks a merchant vessel</td>
</tr>
<tr>
<td>P Hijack prob. ρu, ρa</td>
<td>[0, 1]</td>
<td>Probability of hijack of (un)aware merchant vessel</td>
</tr>
<tr>
<td>N Helicopter</td>
<td>Y/N</td>
<td>Presence of helicopter on board the navy vessel</td>
</tr>
<tr>
<td>N Action radius</td>
<td>[100, 200] nm</td>
<td>Navy vessel distress call response radius</td>
</tr>
<tr>
<td>N Helicopter speed</td>
<td>[140, 170] kn</td>
<td>Speed of navy vessel’s on-board helicopter</td>
</tr>
<tr>
<td>N Cruising speed</td>
<td>[20, 30] nm</td>
<td>Speed of a navy vessel</td>
</tr>
</tbody>
</table>

attacking unsuccessfully for a period given by pirate’s attack time parameter, the merchant vessel continues its voyage according to the original plan; the pirate vessel recovers from the attack for a specified period of time (cool-down time parameter) and then it looks for another target to attack. If the attack is interrupted by the navy, the pirate is disarmed and sails to its home anchorage without further trying to attack any merchant vessels.

3.7. Piracy Countermeasures Model

Merchant and navy vessels can engage in various piracy countermeasures designed to increase the security of passage through piracy-affected waters. Most of such measures require cooperation between multiple vessels and can be viewed as multi-agent coordination mechanisms that augment standard, single-agent vessel behaviors. Based on discussions with the maritime security community, we support the following operational piracy countermeasures in the AgentC model:

- **Recommended transit corridors**, which concentrate merchant traffic along a defined route connecting a sequence of waypoints. Such concentration of traffic facilitates protection from navy vessels; however, it also makes targeting transiting vessels easier for pirates. The corridors are modeled as extensions of the merchant voyage planner. Case study in Section 6 examines the effectiveness of corridor systems closer.
Figure 6: Sequence diagram of the pirate–merchant–navy vessel interaction during a pirate attack. The attack takes a predefined amount of time and is terminated either by the successful hijack of the merchant vessel or by the arrival of a navy vessel or its helicopter.
Table 5: Piracy countermeasures considered, with sets of parameters by which they are specified.

<table>
<thead>
<tr>
<th>Countermeasure</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transit corridor</td>
<td>Sequence of GPS waypoints</td>
</tr>
<tr>
<td>Navy vessel deployment</td>
<td>Set of locations</td>
</tr>
<tr>
<td>Group transit scheme</td>
<td>corridor, Speed levels, transit schedule (per speed level)</td>
</tr>
<tr>
<td>On-board security team</td>
<td>Alertness of merchant vessels</td>
</tr>
</tbody>
</table>

- **Group transit schemes**, which coordinate the timing of merchant vessel transit so that vessels pass high-risk piracy areas in groups. This improves mutual awareness and facilitates navy response; however, it makes the transit take longer as vessels have to follow a predefined schedule and may have to reduce their cruising speeds to match the speed of their respective transit group. The group transit schemes are modeled as an extension of the voyage planner; they assign time marks to a subset of waypoints in a plan and the merchant vessels is then required to be at those waypoints at given time.

- **Navy vessel deployments**, which deploy navy vessels in strategic locations from where they can provide assistance to nearby merchant vessels in case of a pirate attack. We consider only stationary deployments (see Section 3.4).

- **On-board security teams** consists in deploying armed security personnel onboard of vessels transiting high-risk areas capable of deterring attackers and denying them access to the vessel. This countermeasure is currently modeled by the alertness and awareness parameters of the merchant vessel.

Each countermeasure is parameterized by a set of parameters (see Table 5). Except for route randomization, all above measures are currently actively used, although convoy schemes are operated rather sporadically by national navies on an ad-hoc basis. The usage of transit corridors and group transits is currently limited to the Gulf of Aden.

3.8. Simulation Model Outputs

By its very nature, the agent-based micro-simulation model allows recording and evaluating, at different levels, multitude of information about the behavior of the modeled maritime transportation system.

At the lowest level, each simulation run produces a detailed log of all events generated by vessel agents and their interactions among themselves and with the modeled maritime environment. One year of simulated maritime traffic generates hundreds of thousands of events recording vessel locations in time, state-transition events (e.g., `destination-reaching` event, `target-area-selected` event etc.) and events generated during vessel interactions (e.g., `pirate-spotted` event, `pirate-attack` event, `pirate-attack-disrupted` event and many others). Events logs
can be used for studying micro-level behaviors involving one or more individual vessels. They can be also used for on-line visualization of individual simulation runs, which is crucial for presentation and face validation purposes.

Micro-level event logs are aggregated in time and/or space into meso-level spatio-temporal output reports describing the occurrence of specific event or a set of events over time or in geographical space. A particularly important type of output report at this level are *density maps* which capture the frequency of occurrence of a specific event in grid-discretized space (as an example, see Figure 9a for a density map of pirate attacks). Meso-level reports are useful for understanding how a certain phenomena is geographically distributed, how it is changing in time or both. It can also be used for calibration and validation of the model.

Finally, at the highest-level, event logs and spatio-temporal reports are aggregated into simple numerical statistics summarizing the activity in the piracy-affected waters. Both security-related and operational output quantities are evaluated. The former includes *pirate attack count*, i.e, the number of all attacks over a given time period (e.g., a year), and *attack success ratio*, i.e, the number of successful attacks (i.e. hijacks) divided by the number of all attacks; the later includes *average transit distance* and *average transit duration*, which can be used to estimate operational shipping costs. The high-level statistics are used to gain insight into the global behavior of the modeled system under different circumstances and are also crucial for model calibration.

4. Model Implementation

Agent-based maritime transportation simulation requires agent control architecture capable of expressing required individual and collective vessel behaviors. Vessel agents have to be able to execute long-running actions while reacting to interruptions. The minimum intelligent agent architecture that can handle such requirements is the model-based reflex agent architecture (Russell et al., 2010) with encapsulated deliberative modules handling route-planning and other complex reasoning tasks. The required class of behaviors should be implementable in a modular and extensible way, facilitating sharing of common behavior fragments between different classes of vessels. At the same time, the agent control architecture should be computationally efficient enough to handle thousands of simulated agents. Unfortunately, none of existing agent architectures or simulation platforms supports these requirements. AnyLogic\textsuperscript{10} simulation software comes closest but is not suitable due to its commercial closed-source nature. We have therefore implemented our own agent architecture for executing individual agent behavior models.

The implemented architecture decomposes agent behaviors into individual activities (depicted as boxes in Figures 3, 4, and 5) which correspond to the principal activities of the vessel agent (such as cruise, board, patrol etc.) and

\textsuperscript{10}AnyLogic http://www.xjtek.com/
Figure 7: Screenshot of the AgentC simulation. Google Earth-based visualization of the simulation execution and graphical user interface for simulation control and inspection of internal state of the model.

their associated actions. Each activity stores its context when deactivated so that when reactivated, the context can be restored to continue the previously interrupted action. The transition between activities is conditioned by external (e.g., request for help) or internal events (e.g., cool-down time passed). Although limited (e.g., not capable of executing concurrent activities), this approach provides a good trade-off between expressiveness, modularity and computational efficiency.

Behavioral models are embedded and executed in a Java-based agent-based simulator built partially using the lightweight Alite (Novák et al., 2012) multi-agent simulation toolkit and employing Google Earth for geo-spatial visualization (see Figure 7). The simulator provides suitable abstractions for implementing the model of the maritime environment, environment-to-agent sensor interfaces and agent-to-agent communication protocols. Time-stepped simulation execution model is used although migration to the discrete event-based model is possible.

5. Model Calibration and Validation

The AgentC model contains a wide range of parameters that needs to be specified. Most of these parameters were set based on consulting domain sources and experts. There are, however, also parameters which significantly affect the behavior of the model and for which no reliable sources exist—the values of these parameters were therefore determined through calibration against real-world data.
In the following sections, we describe the calibration and validation process employed. Specifically we describe the calibration methodology selected, sensitivity analysis, calibration of the merchant traffic sub-model, and calibration and validation of the complete model.

5.1. Calibration Methodology

The purpose of the calibration step is to set the values of key model parameters so that the behavior of the model most closely reflects the behavior of the real system; this closeness of the model is measured in terms of several fitness criteria. Due to limited supply of computing resources, we performed greedy iterative calibration—in each iteration, we chose a subset of parameters most influencing the fitness criteria, found the optimal value of these parameters and fixed them in subsequent iterations. To further speed up calibration, we used different calibration fitness metric in each step. The ordering of steps and the choice of calibrated variables and fitness metrics in each step was based on the results of the sensitivity analysis. Depending on the standard deviation of the selected fitness criterion, we executed between 50 and 100 simulation runs (with a different random generator seed) for each model configuration (i.e., each combination of model parameters—see Table ??).

In order to compare the spatial outputs of the model, we used spatial success rate (SR) curves as described by Chung and Fabbri (2003). Spatial success rate curves give a concise account of the performance of a spatial model. Specifically, the curve specifies what percentage of space a given spatial model needs in order to cover a given percentage of real-world occurrences of the event of interest (e.g., pirate attacks). The smaller the area required to cover a given percentage of the events, the tighter the fit and the better the model. The SR curve is constructed by discretizing the modeled area into a finite-size cells and then sorting the cells according to the relative frequency of the event occurring in the cell (e.g., relative frequency of pirate attacks). The SR percentage value for x percent of covering cells is determined by taking the x percent of the highest-frequency cells and counting the percentage of events occurring within these cells. SR curves can be modified in order to be used for comparing spatial models against spatial event density maps rather than sets of discrete events. In this case, coverage is counted as the fraction of the overall mass11 of the density map covered by a given proportion of highest-ranking cells. We further define the SR curve index as the percentage of the area under the SR curve with respect to the overall rectangular plot area (i.e., the area below and above the curve). The interpretation of SR curves is illustrated in the captions of Figures 8 and 9. To our knowledge, this is the first use of SR curves for calibration of simulation models.

11Where the mass of a density map is the sum of values in all cells. Note that we assume the space is discretized into finite-sized cells, e.g., a latitude-longitude grid utilized for voyage planning or navy vessel location assignment (see Sections 3.3 and 3.4).
Table 6: Coefficients of variation for each criteria and model parameter. The bold-faced values correspond to parameters which were varied when calibrating the model for each criterion.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Attack Dist.</th>
<th>Attack Freq.</th>
<th>Hijack Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>#N</td>
<td>0.15</td>
<td>0.24</td>
<td>0.32</td>
</tr>
<tr>
<td>#P</td>
<td>0.24</td>
<td>0.74</td>
<td>0.041</td>
</tr>
<tr>
<td>P Visibility radius</td>
<td>0.052</td>
<td>0.26</td>
<td>0.11</td>
</tr>
<tr>
<td>M Alertness</td>
<td>0.053</td>
<td>0.075</td>
<td>0.20</td>
</tr>
<tr>
<td>P Hijack prob. ρₐ, ρᵤ</td>
<td>0.057</td>
<td>0.078</td>
<td>0.16</td>
</tr>
<tr>
<td>P Navy knowledge</td>
<td>0.1</td>
<td>0.085</td>
<td>0.14</td>
</tr>
</tbody>
</table>

5.2. Sensitivity Analysis

Before the actual calibration, we performed sensitivity analysis in order to understand how the variation of key model parameters affects model outputs: (1) attack distribution, (2) attack frequency and (3) attack success ratio. Table 6 summarizes the sensitivity of each output to the variation of the given model parameter, measured in terms of the coefficient of variation\(^\text{12}\) of a given output variable when varying a given model parameter. For each of the model outputs, we have selected the most sensitive parameters which were then varied while the rest of the parameters remained fixed.

5.3. Merchant Traffic Sub-Model Calibration

In the first calibration step, we calibrated the merchant traffic sub-model of the AgentC model, i.e., the model consisting solely of merchant vessels. The calibration involved estimating all entries in the O-D matrix (see Section 3.3) together with the risk aversion parameter \(α \in [0, 1]\) (see Section 3.3) such that we minimize a difference between the simulated and real-world merchant traffic. We used the SR curve index between the simulated merchant traffic density map in form of the latitude-longitude grid \(G^s_m\) and reference 2011 merchant traffic density map grid \(G^A_m\) provided by AMVER (see Figure 8b) as the calibration fitness metric in this step. The simulated merchant traffic density map \(G^s_m\) is obtained in the following way: we discretize the observed area into a latitude-longitude grid with cell size equal to 1\(^°\), similarly to the AMVER grid. We let the simulation run for one simulated year and for each cell, we record the number of transits by any ship. Note that the two maps do not have to be normalized for the computation of the SR index. We used a canonical piracy risk function modeling the piracy risk as a time-independent function of the distance from main pirate anchorages for risk-aware routing.

In the calibration of the O-D matrix, we search for a number of merchant vessels sailing between any two of 20 major world ports. We assume, that the traffic is symmetric, i.e. the flow between two locations is the same in both directions (the O-D matrix is symmetric) and no vessels sail from one location

\(^{12}\)Coefficient of variation is defined as a ratio of standard deviation and mean: \(c_v = \frac{σ}{μ}\).
to the same location (the diagonal values are zero); i.e., we can consider only the upper triangular matrix excluding the diagonal. The O-D matrix contains 20 rows and columns, resulting into 180 parameters which are interdependent and together with the non-linear fitness criterion (the SR index) pose a difficult problem to be solved optimally. We thus use local search with restarts to estimate the value of entries in the O-D matrix.

The algorithm is listed in Algorithm 2. The O-D matrix is initialized as a zero matrix. All entries from the upper triangular matrix excluding the diagonal (denoted as $U(\text{ODmatrix})$ on the 7-th line) are selected into a list $\text{entries}$, which is randomly shuffled. Then, for each entry from $\text{entries}$, the algorithm tries to increment the entry’s value by a step as long as it leads to the improvement of the SR index. If it is not possible to improve the SR index by incrementing any of the entries above a predefined max value, the algorithm tries to improve the SR index by decrementing the entries by step (switching to decrements on 25-th line). The SR index is computed in each step by simulating one year of merchant traffic sampled from the O-D matrix, creating a merchant traffic density map and comparing it to the reference map.

The O-D matrix calibration was repeated 100 times for each risk aversion coefficient $\alpha = \{0, 0.1, \ldots, 1\}$. The best fit was achieved for $\alpha^* = 0.6$; the resulting traffic density map for $\alpha^*$ and the associated SR curve are given in Figures 8a and 8c, respectively. Minor discrepancies can be observed around Kenyan and Tanzanian coast and in the Mozambique channel; overall, the fit is very good.

5.4. Complete Model Calibration

Following the calibration of merchant traffic sub-model, we calibrated the complete model containing all three categories of vessels. The complete model
Algorithm 2 O-D matrix Calibration

1: input: max, step
2: SRindex ← ∞
3: refMap ← AMVER density map
4: ODmatrix ← U(zeros)
5: best ← 0
6: changed ← true
7: entries ← shuffle(U(ODmatrix))
8: while changed do
9:    changed ← false
10:   for entry ∈ entries do
11:       while (entry ≥ 0)&(entry ≤ max) do
12:          entry ← entry + step
13:       simMap ← simulate(ODmatrix)
14:       SRindex ← getFit(refMap, simMap)
15:       if SRindex < best then
16:          entry ← entry − step
17:          break
18:       else
19:          best ← SRindex
20:          changed ← true
21:       end if
22:   end for
23: if step > 0 then
24:    step ← −step
25: changed ← true
26: end if
27: end while

was calibrated to fit the situation in the Indian Ocean in 2011, where there were 181 attacks (source: IMB 2011 reports), from which 28 were hijacks (15.4% hijack success rate). Even though some of the attacks are unreported and thus the IMB 2011 reports are incomplete, it is to our best knowledge the most comprehensive report source. The calibration consisted of the following three steps:

5.4.1. Attack Spatial Distribution Fitting

First, the complete model was calibrated with regards to the spatial distribution of pirate attacks. The number of navy vessels $\#N$ (0 − 500) and pirate’s $P$-navy-knowledge were the calibrated variables; the SR curve index (see Section 5.1) between the attack density map produced by the model and the IMB 2011 reports was used as the fitness metrics. The best fit was found for $\#N = 50$ and $P$-navy-knowledge = 0.4. The attack density map produced by the model and the SR curve for the best values of calibrated parameters are
Figure 9: Complete model calibration—attack spatial distribution fitting. (a) Density map for the complete model. (b) IMB 2011 Reports. (c) SR curves for the complete model (blue solid line) and IMB 2011 reports (red dashed line). The red SR curve for IMB 2011 reports was measured by transforming the incidents into a density map on which the SR curve was measured. This IMB density map thus serves as a theoretical upper-bound 20% of the most dense cells in the IMB density map covers approximately 72% and 20% of the AgentC model density map covers 61%.

given in Figure 9, along with the reference IMB 2011 reports.

Two directly observable discrepancies between the AgentC model and the situation in 2011 can be observed: the attacks in the AgentC model are concentrated in the East Arabian sea, not spreading to the North. This difference is caused by the fact that the AgentC pirates sailing to the Northern Arabian sea encounter a merchant vessel during their voyage prior reaching the target area and decide to attack this vessel instead of continuing along its original route. Additionally, due to the sparse AgentC merchant traffic in the West Indian Ocean, there are no simulated attacks in that area, the AgentC pirates shift their presence under the southern tip of India, causing another small discrepancy. Even though the fit of the calibrated model on the available data is good, the described discrepancies have to be taken into account when using the model for decision making.

5.4.2. Pirate Attack Frequency Fitting

Second, the complete model was calibrated with regards to the overall number of attacks. The number of pirates $\#P$ (0–5) and pirate’s $P\text{-visibility-radius}$ (5–12 nm) were the calibrated parameters; the fitness metric was a difference between the overall number of attacks produced by the model and the reference real-word value of 181 based on the IMB 2011 reports. The best fit was obtained for $\#P = 2$ and $P\text{-visibility-radius} = 6$ nm, producing on average 182 attacks with the standard deviation of 16.1.

5.4.3. Pirate Attack Success Ratio Fitting

Finally, the complete model was calibrated with regards to the attack success ratio. Two parameters influencing the outcome of the merchant vessel-pirate interaction were calibrated: $M\text{-alertness} \in [0, 1]$ and $P\text{-base-hijack-probability}$
\[ \rho_a, \rho_u \in [0, 1] \] (defined in Section 3.6); the fitness metric was the difference to the reference success ratio based on IMB 2011 report statistics, was 0.15. Best fit was obtained for \( M_{\text{alertness}} = 0.5, \rho_a = 0.2 \) and \( \rho_u = 0.5 \), estimating the probability of a hijack with navy vessels (\#\( N = 50 \)—fixed in the previous calibration phase) to be 0.15 and without any navy vessels to be \( p = 0.35 \). The probability of a pirate being disrupted by a navy vessel is then 43%, according to our model—this is an example of an insight which cannot be directly inferred from the collected attack reports alone.

5.5. Validation

Face validation was performed repeatedly throughout the model development process. We consulted experts and officials from the industry, government and military, including International Maritime Organization, U.S. Naval Research Lab, U.S. Naval Postgraduate School and several maritime security providers. Feedback received on structural walkthroughs and visualized simulation runs confirmed structural and behavioral plausibility of the proposed model.

Unfortunately, due to lack of data on the behavior of the maritime transportation system under varying circumstances (e.g., the exact number of pirate attack groups and/or deployed naval warships), we were unable to statistically validate the model. The model should not therefore be treated as reliable for quantitative prediction and should be used for gaining qualitative insights only.

6. Case Study

We have applied the developed model to several real-world use cases, based in part on discussions with maritime domain stakeholders. Here we present one particular case study focusing on analyzing the possibility of introducing transit corridor system in the Indian Ocean.

The existing International Recommended Transit Corridor (IRTC), established in 2009, has proven—in combination with the deployment of navy vessels—a very effective tool in reducing the number of successful pirates attacks in the Gulf of Aden. The maritime security community has been discussing the possibility of establishing additional corridors in the Indian Ocean, where most pirate activity takes place following pirates’ displacement from the Gulf of Aden. In contrast to the Gulf of Aden, which is an elongated, narrow area with a simple bidirectional traffic flow, the Indian Ocean is much larger and crisscrossed, in all directions, by a multitude of traffic flows. This makes the design of an effective corridor system a complicated task.

6.1. Scenarios

We studied the effect of two possible layouts of Indian Ocean corridor systems: (1) single west-east corridor channeling the large amount of west- and east-bound traffic (denoted as Single-IO), and (2) a more extensive multi-corridor system covering all the main traffic flows in the Indian Ocean (denoted
Figure 10: Corridor layouts for the Indian Ocean corridor system. The Single-IO layout only uses IRTC with the red east-west corridor; the Multi-IO layout utilizes all depicted corridors.

as Multi-IO). See Figure 10 for a scheme of corridor layouts. We compared the results with the current setup where no corridors are used in the Indian Ocean (denoted as None-IO). The existing IRTC corridor was considered in all three configurations.

In addition to the corridor layout, we were interested in assessing synergies between corridors and other countermeasures, specifically in assessing employing group transit schemes within the corridors and deploying of navy vessels alongside corridors. In addition to the corridor layout, we therefore included the number of deployed navy warships (\(N = \{20, 30, 40, 50, 60, 80, 100\}\) and the use of group transit (\(\text{group-transit} = \{\text{YES, NO}\}\)) as additional study parameters. In order to make the assessment more robust with respect to the variation of future pirate activity, we also included the number of active pirates (\(P = \{1, 2, 3, 4\}\)) as a study parameter.

6.2. Results

The results given are for one year of simulated maritime traffic. Due to probabilistic nature of part of the model, we simulated each configuration for 50 runs and present average values together with standard errors.

The values of the average transit distance (in nautical miles) and average transit duration (in hours) only depend on the layout of the corridor system and amounted to 2153 nm / 141 h for the None-IO setup with no corridors in the Indian Ocean, 2162 nm / 142 h for the Single-IO and 2213 nm / 145 h for the Multi-IO corridor setups. The small difference between different corridor settings is due to the positioning of corridors copying main natural shipping lanes. The traffic is not re-routed significantly by the introduction of the corridors. Although some of the routes are longer, others can actually be shorter—the risk inside corridors is considered zero\(^{13}\) and the corridors (especially the south-north corridor) can therefore act as shortcuts to long risk-avoiding detours.

\(^{13}\)Estimating the value of risk inside a corridor opens another case study by itself—what is the actual risk in an established corridor, given its positioning and a specified number of patrols? Are the merchant vessels incentivized enough to transit through the corridor? AgentC can be used to estimate the risk value by repeatedly running the simulation, compare the
Figure 11a captures the dependency of the number of hijacks on the number of navy vessels for each corridor system, averaged over different numbers of pirates. As expected, increasing the number of navy vessels decreases the number of both attempted and successful attacks. The reduction in attempted attacks is caused by a denser navy presence, which causes the pirates not to launch attacks when a navy vessel is nearby. The reduction in successful attacks is then caused, additionally, by more frequent attack disruption allowed by the higher density of navy vessels—this can be seen from Figure 11b which depicts a detailed breakdown of attack outcomes for the Multi-IO corridor system. When the number of navy vessels is increased to a certain level, most of the attacks are successfully disrupted.

What is more interesting is the finding that in order to have positive impact on reducing hijacks, the extended corridor systems (Single-IO and Multi-IO) have to be patrolled by a high number of navy vessels (approximately 70 and more). For fewer than 40 navy vessels, the introduction of the corridors in the Indian Ocean actually worsens transit security (keep in mind that, as suggested in the validation section, quantitative results are only indicative). This is because the better predictability and higher concentration of merchant traffic inside the extended corridors systems makes targeting vessels easier for pirates. This can be seen from Figure 11c—the number of attempted attacks for the Multi-IO remains higher than for the None-IO setup even for very high numbers of navy vessels. The ratio of disrupted attacks to the number of all attacks (depicted in Figure 11d) can be seen as navy vessel efficiency. This efficiency rises with increasing the number of deployed navy vessels. More interestingly, the Single-IO and Multi-IO systems lead to higher navy vessel efficiency (though this increase is not enough to counter the increase of attempted attacks). The use / not use of group transit has insignificant effect in the current model—this may change if more sophisticated patrolling strategies which coordinate navy vessels with transit groups are employed.

Finally, in Figure 12 we compare the geo-spatial distribution of vessel hijacks between None-IO and Multi-IO corridor setups for 50 navy vessels. The distribution of attacks in the Multi-IO corridor system is more concentrated along the corridors and the pirate activity is shifted from the Gulf of Aden and from the East Arabian Sea to the corridor along the Somali coast.

Overall, the results suggests that the positive effect of transit corridors is not directly transferable from the small and narrow Gulf of Aden into the vast Indian Ocean. This is not surprising given the complex nature of the interdependencies in the maritime transportation system; it is exactly the kind of conclusions that is difficult to reach without in-depth simulation modeling (e.g., by employing data analysis techniques only).

Key limitation of the study lies in the simple static deployment of navy ves-

---

number of incidents inside and outside the corridor, quantify the pirate target area preferences, assess the risk and run another simulation with modified risk values, until convergence. Unfortunately, this study is out of scope of this paper.
Figure 11: Results of the Corridor system study. (a) Dependency of the number of hijacks on the corridor system and the number of navy vessels (lower is better). (b) Breakdown of the different attack types for Multi-IO corridor system. (c) Dependency of the number of all attacks on the corridor system (lower is better). (d) Dependency of the ratio of intercepted attacks on the number of navy vessels—we can observe boost of navy vessel efficiency by the introduction of extended corridor systems, when enough navy vessels are available (higher is better).

sels. More elaborated patrolling and convoy formation strategies could be more effective and allow the extended corridor systems to be successfully patrolled with fewer vessels. The agent-based design and implementation of the simulator makes introduction of such strategies into the model straightforward assuming the description of the strategies can be obtained.

7. Conclusions

We presented AGENTC, a simulation model of the maritime transportation system affected by piracy. The model employs agent-based modeling approach—the behavior of the overall system is represented as a composition of thousands of micro-level behaviors of individually simulated vessels. To our knowledge, AGENTC is the first model representing deep sea shipping and pirate activity at such a level of detail.
The ability of the model to provide insight into the complex dynamics of piracy-affected waters was demonstrated on a real-world use case of designing transit corridors in the Indian Ocean. Although direct extrapolation of the experience from the Gulf of Aden would suggest that corridors will boost transit security, the simulation of several corridor layouts revealed that this is not necessarily the case and that additional factors play a decisive role. Many other policy decisions can be analyzed using the AgentC model, either out of the box or after small extensions. In fact, applications of the model are not limited to maritime piracy—the merchant traffic sub-model alone is a valuable result and can be used for studying the impact of other factors on maritime shipping (e.g., fuel costs, opening of northern shipping lines etc.).

The major obstacle to building the model was a severe lack of data on almost all aspects of the behavior of the maritime transportation system. This made proper statistical validation of the model impossible and, consequently, all conclusions have to be interpreted with caution. On the positive side, the work on the model allowed us to clearly identify missing datasets and such information can now serve as a motivation and direction for future maritime data acquisition activities.

Although the AgentC model is the main contribution, the methodology using which the model was developed and calibrated is also a valuable contribution. The application of the agent-based simulation engineering process in the maritime security domain is novel and required several iterations to refine. The trialed-and-tested methodology can be enacted repeatedly to model other maritime transportation scenarios or to improve the existing model when new data become available.
Acknowledgements

This research is funded by the Office of Naval Research (grant no. N000140910537) and by the Czech Ministry of Education, Youth and Sports (grant no. LH11051).

References


