

Assessing Scene Generation Techniques for Testing COLREGS-Compliance of Autonomous Surface Vehicles

DOMINIK FREY, Linköping University, Sweden

ULF KARGÉN, Linköping University, Sweden

DÁNIEL VARRÓ, Linköping University, Sweden and McGill University, Canada

Autonomous surface vehicles (ASVs) need to complete missions without posing risks to other maritime traffic. Safe traffic in open sea encounters is controlled by the International Regulations for Preventing Collisions at Sea (COLREGS) formulated by the International Maritime Organization (IMO). Designed with human operators in mind, the COLREGS are intentionally underspecified, which may result in ambiguous requirements for correct behaviour for ASVs. Hence the systematic testing of such ambiguous situations is particularly important. This paper investigates to what extent existing test scenario generation approaches deemed effective in the automotive domain can be adapted to test COLREGS-compliance in a maritime context with multi-vessel encounters. In a series of experiments involving synthetic and real-world test scenarios, their performance is evaluated with respect to relevance, diversity, completeness, scalability and speed. Our results indicate that (1) test scenarios derived from historic maritime traffic are insufficient for testing multi-ship encounters. Moreover, (2) existing test scenario generation techniques provide sufficient scalability and speed, but they are very limited in terms of diversity and completeness when the number of vessels increases.

CCS Concepts: • **Computing methodologies** → **Model development and analysis**; • **Software and its engineering** → **Software testing and debugging**.

Additional Key Words and Phrases: autonomous surface vehicles, COLREGS, functional scenarios, logical scenarios, test scene generation, genetic algorithms, rejection sampling

ACM Reference Format:

Dominik Frey, Ulf Kargén, and Dániel Varró. 2025. Assessing Scene Generation Techniques for Testing COLREGS-Compliance of Autonomous Surface Vehicles. *Proc. ACM Softw. Eng.* 2, ISSTA, Article ISSTA044 (July 2025), 24 pages. <https://doi.org/10.1145/3728919>

1 Introduction

In recent years, significant efforts have been made to develop autonomous surface vehicles (ASVs) for civilian and military purposes. A key challenge in developing ASVs is to ensure that they can navigate safely without endangering other vessels. Specifically, any vessel operating in international waters must comply with the International Regulations for Preventing Collisions at Sea (COLREGS) [19], a set of rules standardized by the International Maritime Organization. COLREGS regulate how a vessel (referred to as *own ship*, or OS) should navigate to avoid a potential collision when it encounters another vessel (the *target ship*, or TS) on a near-intercept course.

Significant effort has been put into developing COLREGS compliant control software (see, for example, [73]), and maritime simulators [2, 72] for full-stack testing of ASVs in virtual environments.

Authors' Contact Information: Dominik Frey, Linköping University, Linköping, Sweden, dominik.frey@liu.se; Ulf Kargén, Linköping University, Linköping, Sweden, ulf.kargen@liu.se; Dániel Varró, Linköping University, Linköping, Sweden and McGill University, Montreal, Canada, daniel.varro@liu.se.



This work is licensed under a Creative Commons Attribution 4.0 International License.

© 2025 Copyright held by the owner/author(s).

ACM 2994-970X/2025/7-ARTISSTA044

<https://doi.org/10.1145/3728919>

However, there is currently a lack of effective means for *systematic validation* of COLREGS compliance. Existing work either relies on manually designed test scenarios [10, 47, 53, 68, 79] or scenarios extracted from historical traffic data [5, 40, 43, 46, 76, 85]. Unfortunately, none of these approaches offer a proper solution for the safety assurance of ASVs, as the former does not scale to scenarios with a large number of vessels, and the latter is unlikely to cover rare critical scenarios.

Although there is a significant body of work on synthesizing high-quality test scenarios for autonomous road vehicles [1, 3, 6, 8, 13, 27, 34, 54, 75], the maritime domain is highly different, hence these techniques are not directly applicable for testing COLREGS compliance. Particular challenges posed by the maritime domain compared to the automotive domain include:

- *Completely different vessel dynamics.* Compared to road vehicles, the physical interactions between actors and the environment are more complex in the maritime domain, where vessel dynamics are influenced by the shape of the hull, and external forces such as waves, currents and wind. Moreover, the visibility range is of significant maritime concern, as distances between actors are several orders of magnitude greater than between road vehicles.
- *Less constrained geometry.* The spatial relations of actors in the automotive domain are restricted by road geometry, which is in turn restricted to adhere to a finite set of road design elements (intersections, roundabouts, lanes, etc.) and traffic rules. This makes the search space of possible physical actor relations comparably much smaller than in the maritime domain, where actors can have any relative position, heading, and speed. The higher degrees of freedom makes, among other things, stochastic testing (which is widely used in the automotive domain [36, 82]) largely infeasible for testing COLREGS compliance.
- *More diverse classes of actors.* The size difference of actors (hence, their manoeuvring capabilities) is at most 2-3 orders of magnitude for road vehicles (e.g., motorcycle vs. semi-truck), while it can be up to 6 orders of magnitude in maritime (e.g., small motor boat vs. oil tanker).
- *Free-form trajectories.* An implication of the two points above is that maritime vessel trajectories are constrained not by a predetermined road geometry, but rather by the COLREGS rules, by the manoeuvrability of the vessel, and by external dynamics (currents, wind, etc.). Therefore, generating a test scenario for the maritime domain involves two principal steps: (1) generation of an *initial scene*, which defines initial actor placements, headings, and speeds; and (2) generation of *vessel trajectories* evolving from the initial scene.
- *Underspecified traffic rules.* Being designed with human operators in mind, the COLREGS are deliberately underspecified, leaving significant room for individual judgement by ship crew when making decisions. As a consequence, the COLREGS can become *ambiguous* for multi-ship encounters, i.e., COLREGS compliant behaviour w.r.t. one vessel can lead to a rule violation (and risk of collision) w.r.t. another vessel.

Based on this, and further interactions with maritime experts in an ongoing project, we can formulate the following desiderata for an ideal maritime test scenario generator:

- D1 **Relevance.** Each test case should represent a hazardous situation that may lead to a potential collision (or other safety violations) if the OS fails to take appropriate action. While relevance is difficult to quantify, according to [41], ambiguous COLREGS situations represent a particularly relevant subclass of test scenarios as identified by maritime experts.
- D2 **Diversity.** An ideal scenario generator should produce *semantically different* test cases to avoid the redundant testing of equivalent COLREGS scenarios. This is especially important if test scenarios are executed in computationally expensive, high-fidelity simulators.
- D3 **Completeness.** Related to the notion of diversity, an ideal scenario generator should enumerate *all semantically distinct COLREGS scenarios* for a given number of vessels. In particular,

for safety assurance, it is important to cover not only common types of encounters, but also rare safety-critical edge cases (e.g. ambiguous COLREGS situations).

D4 Scalability. To cover more complex scenarios, such as those exhibiting ambiguity, the scenario generator must be capable of scaling to a larger number of vessels.

D5 Speed. Since a large number of scenarios might be needed to cover the space of possible COLREGS situations, the time for generating a single scenario is also an important factor.

In this experience report, we investigate existing state-of-the-art scenario generation approaches from the automotive domain, and evaluate to what degree they are applicable to the maritime domain with regards to **D1–D5** above. In particular,

- We formalize the (informally presented) COLREGS model proposed in [41], and extend it to handle an arbitrary number of actors (i.e., vessels) for open sea encounters.
- We formally define logical scenarios and initial scenes for open sea encounters in the maritime domain with their geometric constraints (analogously to logical scenes in multi-level abstractions for scene concretization in the automotive domain [3]).
- We adapt two scenario generation approaches widely used in the automotive domain (*search-based scenario generation* (**SB**) [1, 6, 8, 34], and *rejection-sampling*, used by the state-of-the-art Scenic tool (**RS**) [27, 75, 81]) to generate initial scenes in the maritime domain.
- In a series of experiments involving the automated generation of initial scenes, we evaluate the two approaches with respect to **D1–D5**. We also compare the synthesized scenes with CommonOcean (**CO**) [43], a recently published dataset of historical maritime traffic collected using public AIS (automatic identification system) data.

In this paper, we exclusively focus on the *generation of initial scenes*, which is the conceptually hard problem in scenario generation for the maritime domain. Subsequently, concrete trajectories of vessels can be generated from the initial scenes by various approaches [24, 57, 70]. The scope of our investigations is limited to *open sea encounters involving multiple vessels*.

Up to our best knowledge, our paper provides the first cross-domain adaptation of test scenario generation techniques developed for automotive in the maritime domain. Moreover, it provides a systematic evaluation of various key quality attributes (derived with the involvement of maritime experts) on both synthetic and real-world test scenarios. Moreover, testing COLREGS-compliance in multi-vessel open sea encounters is another innovative aspect of our work.

2 Maritime Background

2.1 COLREGS Concepts

The COLREGS contain 41 rules to ensure safe maritime traffic in open seas. While ports involve often busy maritime traffic, only a limited number of vessels (typically 2-6) are involved in open sea encounters [43, 76, 85]. Moreover, in busy waters close to shorelines, local traffic separation schemes, which override COLREGS, are often used. Rules 1–19, covering steering and sailing, and Rules 21–22, focusing on light placement, shapes, and visibility requirements, are the most relevant for evaluating the safety of ASV control systems. The remaining rules primarily cover administrative and operational details concerning lighting and shapes for specific vessel types and specialized situations. This section discusses the rules relevant for the scope of this study.

- **Rule 6: Safe speed:** Vessels must maintain a "safe speed" at all times, but COLREGS offer limited guidance on this. While moderating speed is critical in restricted visibility (e.g., fog) [14], in good visibility and open sea conditions, "safe speed" is typically the vessel's cruising speed.

Table 1. Beam angles for identifying relative bearing sectors.

$\angle_{\text{bow}}$	10°
$\angle_{\text{side}}$	107.5°
$\angle_{\text{stern}}$	135°
$\angle_{\text{masthead}}$	225°

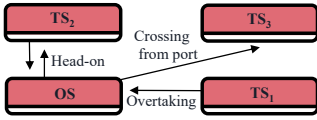


Fig. 1. Sample COLREGS relations between OS, TS₁, TS₂, TS₃.

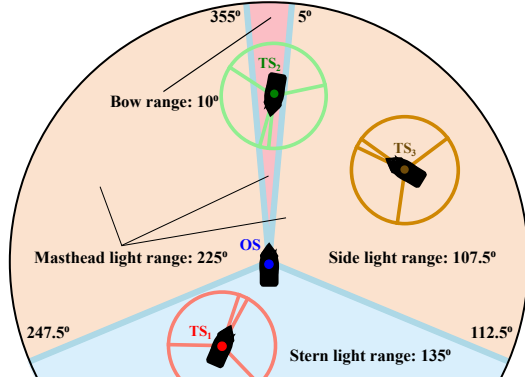


Fig. 2. COLREGS situations and sectors based on signal light types and beam angles. TS₁ is overtaking OS, OS and TS₂ are head-on, OS is crossing TS₃ from port.

- **Rule 7: Risk of collision:** A *course* is defined by the vessel's constant speed and heading. According to the COLREGS, a risk of collision exists when two vessels are on a *collision course*, i.e., without any intervention, they will likely collide in the future.
- **Rule 8: Action to avoid collision:** This rule emphasizes maintaining a "safe distance" and making "readily apparent" course changes. A safe distance generally means a significant enough distance between the vessels so that hydrodynamic effects cannot lead to contact. It is often approximated by a convex area around the vessel called the *safety domain*. Some commonly used elliptical safety domains are mentioned in [15, 18, 60, 69, 77]. For course changes to be "readily apparent", common practice based on case law and custom [78] suggests a change of at least 30° .
- **Rule 11: Deemed COLREGS situations:** Rule 11 indicates that the specific positions and velocities determine each ship's collision avoidance responsibilities when the vessels first get within the *visibility range* [35]. These responsibilities are detailed by Rules 12-18 (see Section 2.2).
- **Rule 22: Visibility range:** The visibility range of a vessel is usually between 1 and 6 nautical miles (NM) based on the required range and beam angle of positional lights, depending on ship size.

2.2 Defining COLREGS Situations and Scenarios

We define a COLREGS *situation* as a relationship between two vessels based on their relative bearing. In a COLREGS situation, the vessels must be within each other's visible range, and a risk of collision exists. We refer to such situations that arise in a multi-vessel encounter as a COLREGS *scenario*.

Rules 13, 14, and 15 identify three types of COLREGS situations based on the *relative bearing sectors* around the vessels. These bearing sectors originate from the visibility requirements of signal lights (position, beam angle), as specified in COLREGS Rule 21. The sectors and their angular ranges, as noted in Table 1, are illustrated in Figure 2 with a 4-vessel scenario between the *own ship* (OS), and *target ships* (TS₁, TS₂ and TS₃). This same sector classification is commonly used in related literature [38, 76]. The relation types are listed in Figure 1 based on the scenario in Figure 2:

- **Giving-way and standing-on (Rules 16-17):** A give-way vessel must take substantial early action to avoid a collision course, while a stand-on vessel should maintain its course unless the give-way vessel does not satisfy its obligations.

- *Overtaking (Rule 13)*: **OS** is overtaken by **TS₁**, if **TS₁** is within the stern light range ($\angle_{stern}$) of **OS**, and **OS** is within the masthead light range ($\angle_{masthead}$) of **TS₁**. In this situation, **OS** has to give way to **TS₁**, while **TS₁** stands on its course.
- *Head-on (Rule 14)*: **OS** is head-on with **TS₂**, if **TS₂** is within the bow range ($\angle_{bow}$) of **OS**, and **OS** is also within the bow range ($\angle_{bow}$) of **TS₃**. Both **OS** and **TS₂** have give-way responsibilities.
- *Crossing from port (Rule 15)*: **OS** is crossing **TS₃** from port if **OS** is within the left side range ($\angle_{side}$) of **TS₂**, and **TS₃** is within the masthead light range ($\angle_{masthead}$) of **OS**. In this situation, **OS** has to give way to **TS₃**, while **TS₃** stands on its course.

Ambiguity in COLREGS: One fundamental limitation is that Rules 13–15 only address two-vessel encounters, leaving multi-ship interactions underspecified and potentially hazardous [41]. For example, in the scenario shown in Figure 3a, a vessel can be simultaneously in conflicting stand-on and give-way situations, risking collisions despite compliance with the rules.

2.3 Towards Assessing COLREGS Compliance by System-Level Testing

Recent initiatives [59, 70] assessing COLREGS compliance of an ASV in open sea conditions mostly rely on system-level testing in a simulated environment [2, 72]. First, COLREGS scenarios are generated automatically as test cases. Each test case forces the ASV to solve a challenging encounter involving other vessels. Then, each scenario is simulated to assess if ASV adheres to the regulations.

A *test case* consists of (1) an *initial scene* with precise position, heading, and speed for each vessel, and (2) COLREGS-compliant *trajectories for target ships*. A *trajectory* is a sequence of positions and courses for a ship over time. As (3) *test oracle*, one may define properties of acceptable behaviour and trajectories for the ASV under the assumption that all target ships adhere to their predefined trajectories. The synthesis of initial scenes, trajectories and test oracles are regarded as separate challenges for ASV testing. This paper focuses on the generation of initial scenes (1), which has the most significant effect on the overall interaction and safety hazards in a COLREGS scenario [12].

2.4 Multi-Level Abstractions for COLREGS Scenarios

Analogously to previous work on scenario generation for autonomous driving [3, 51], COLREGS compliance can be investigated on multiple abstraction levels in a maritime context [41].

System-level ASV testing is carried out using simulators that execute *concrete scenarios* as test cases consisting of initial scenes, and detailed trajectories for all target ships. The initial scene marks the start of a scenario, typically when two vessels first come into visibility range. Then, the trajectory of each vessel determines its position, heading, and speed at a specific point in time.

An initial scene for an open sea encounter of vessels must satisfy various geometrical and numerical constraints (e.g., relative bearing in a crossing encounter), represented by *logical scenarios*. Each initial scene represents one solution of the set of constraints defined by a logical scenario. In this paper, a logical scenario represents a potentially dangerous multi-vessel COLREGS-regulated encounter, which sets the ASV and one or more target ships on a potential collision course.

For a systematic investigation of semantically different COLREGS situations that involve a certain number of target ships, *functional scenarios* can be derived from logical scenarios by *qualitative abstraction* (QA) [4]. Using qualitative abstraction, one can derive a graph model (based on relational algebra) to represent functional scenarios by abstracting geometrical constraints between vessels into high-level relations such as "crossing from port situation". Next, Section 3 discusses how to model functional COLREGS scenarios formally, while Section 4 precisely defines logical scenarios.

Table 2. Constraints for consistent functional models of COLREGS scenarios $C \subset \mathcal{F}$.

Informal description	Well-formedness constraint ($\forall \langle o, o_i, o_j \rangle \in O_F^3, o_i \neq o_j$)
A vessel cannot be in a COLREGS relation with itself.	$\forall r \in \mathcal{R}_{sit} : \neg r(o_i, o_i)$
If there is a COLREGS situation between two vessels, there cannot be any other COLREGS situation between them.	$\forall \langle r_k, r_l \rangle \in \mathcal{R}_{sit}^2, r_k \neq r_l : (r_k(o_i, o_j) \implies \neg r_l(o_i, o_j))$
Each COLREGS (except for <i>headOn</i>) holds in one direction.	$\forall r \in \mathcal{R}_{sit}, r \neq \text{headOn} : (r(o_i, o_j) \implies \neg r(o_j, o_i))$
The <i>headOn</i> situation must be defined in two directions.	$\text{headOn}(o_i, o_j) \implies \text{headOn}(o_j, o_i)$

3 Functional Modelling of COLREGS Scenarios

Domain-specific graph models are frequently used to model complex cyber-physical systems [31, 51]. In this paper, (1) graph models provide a precise means to define functional scenarios, (2) graph constraints capture COLREGS-specific assumptions, and (3) graph shapes help identify functional-level equivalence classes to address completeness on a high level of abstraction.

Metamodel. An initial metamodel of functional scenarios in maritime was introduced in [41]. An algebraic representation of a metamodel $M := \langle \Sigma, \alpha \rangle$ contains a vocabulary $\Sigma := \mathcal{L} \cup \mathcal{R}_{sit}$, and an arity function $\alpha : \Sigma \rightarrow \mathbb{N}$ (2 for binary predicate symbols, 1 for unary predicates) where

- $\mathcal{L} := \{OS, TS\}$ denotes object roles within the scenario for COLREGS compliance:
 - *OS* denotes the own ship (**OS**) that represents the system under test.
 - *TS* denotes the target vessels (**TS_i**) that form challenging environment for the **OS** vessel.
- $\mathcal{R}_{sit} := \{\text{overtake}, \text{headOn}, \text{cross}\}$ denotes COLREGS situations between two vessels:
 - *overtake*: indicates an overtaking situation between the source and the target vessels.
 - *headOn*: indicates a head-on situation between the source and the target vessels.
 - *cross*: means a crossing from the port situation between the source and target vessels.

Functional models. Functional scenarios are represented as typed graphs where each object has a unique identity. Given a signature $M := \langle \Sigma, \alpha \rangle$ representing the metamodel, a functional scenario model is a logic structure $F := \langle O_F, I_F \rangle \in \mathcal{F}$ where O_F is the finite set of objects in the model, while I_F gives an interpretation of $r \in \Sigma$ over the object set: $I_F(r) : O_F^{\alpha(r)} \rightarrow \{\text{true}, \text{false}\}$.

A *functional model* assigns a truth value to unary relations over objects and binary relations over a pair of objects [4, 29]. For example, in the functional scenario of Figure 3b: $\{I_F(OS)(OS) := \text{true}, I_F(TS)(TS_1) := \text{true}, I_F(TS)(TS_2) := \text{true}, I_F(\text{overtake})(TS_2, OS) := \text{true}, I_F(\text{cross})(OS, TS_1) := \text{true}\}$. We use a relational predicate syntax [65] to query the state of the model with $r \in \Sigma, \hat{o} \in O_F^{\alpha(r)}$:

- $r(\hat{o}) \equiv I_F(r)(\hat{o}) = \text{true}$, i.e. r relation holds between objects in $\hat{o}$.
- $\neg r(\hat{o}) \equiv I_F(r)(\hat{o}) = \text{false}$, i.e. r relation does not hold between objects in $\hat{o}$.

Further constraints of functional scenario models can be captured by a *graph query* q [9], which is equivalent to first-order logic. Each match $m(q)$ of a graph query represents a constraint violation.

Consistent functional scenarios. Even in case of a consistent interpretation, a functional scenario model is *inconsistent* if there is a match $m(q)$ for a graph constraint q in the functional scenario model. Key consistency constraints for (relevant) functional scenarios are formalized in Table 2. The set of *consistent functional models* satisfying with a constraint set $\{q_i\} \in Q$ is denoted as $C \subset \mathcal{F}$.

Relevant test scenarios. In this paper, we aim to derive concrete initial scenes as test cases to assess COLREGS compliance in the context of COLREGS scenarios. Therefore, relevant test cases comprise scenarios that present significant challenges to an ASV while all other ships comply with

Table 3. Constraints for relevant test scenarios $\mathcal{T} \subset \mathcal{F}$.

Informal description	Constraint for test scenarios ($\forall \langle o_i, o_j \rangle \in O_F^3, o_i \neq o_j$)
(1) There is exactly one OS representing an ASV while all other ships are TSs.	$OS(o_i) \implies \neg OS(o_j) \wedge \neg TS(o_i)$
(2) If a vessel is not an OS, then it is a TS.	$\neg OS(o) \implies TS(o)$
(3) Each OS, TS must be in a COLREGS situation.	$OS(o_i) \wedge TS(o_j) \implies \exists r \in \mathcal{R}_{sit} : (r(o_i, o_j) \vee r(o_j, o_i))$

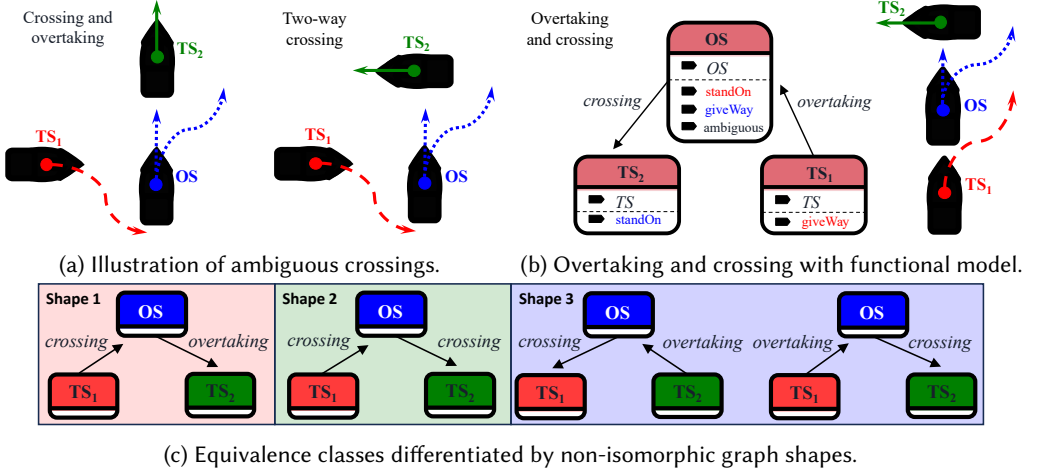


Fig. 3. Examples of functional-level equivalence classes of ambiguous scenes for 3-vessel interactions.

COLREGS rules. We define *relevant test scenarios* $\mathcal{T} \subset \mathcal{F}$, which satisfy the set of constraints listed in Table 3, where $\mathcal{T}$ represents COLREGS scenarios in open sea encounters between (1) a single ASV and (2) a certain number of target ships, (3) involved in a COLREGS situation with each TS which would result in a safety violation without further intervention.

Ambiguous test scenarios. Ambiguous scenarios (where COLREGS compliance w.r.t. one TS results in a COLREGS violation w.r.t. another TS) were identified by maritime experts as a particularly relevant subclass of test scenarios for assuring COLREGS-compliance [41]. The set of *ambiguous functional scenarios* are denoted as $\mathcal{A} \subseteq \mathcal{T}$ after including the following constraint:

The OS must be in an ambiguous situation.	$OS(o) \implies standOn(o) \wedge giveWay(o)$ where $standOn(o) := \exists o_j : cross(o_j, o) \vee overtake(o_j, o)$ $giveWay(o) := \exists o_j : cross(o, o_j) \vee overtake(o, o_j) \vee headOn(o, o_j)$
---	---

For example, Figure 3b shows ambiguous 3-vessel functional scenario models (compliant with $\mathcal{A}$) generated automatically using the Refinery graph solver [49, 50] following recent results [41].

Functional equivalence classes. When testing ASVs, one aims to provide a diverse set of test scenarios that trigger the behaviour of involved vessels in semantically different situations. We aim to characterize the diversity of a test suite by functional(-level) equivalence classes using *structural graph diversity metrics* [66] based on neighbourhood shapes [62]. Informally, such graph shapes categorize all nodes in a graph based on the existence (or non-existence) of specific edges in their neighbourhoods within a given range. While the number of neighbourhood shapes is finite for a given range, enumerating all possible shapes is only feasible for small graphs.

From a testing perspective, the multi-level abstractions [41] (see Section 2.4) ensure that if two functional scenarios have different neighbourhood shapes, then they represent semantically different sets of scenarios. Hence, functional scenario models can be grouped into *functional equivalence classes* characterized by different graph shapes. As a sufficiently diverse test suite needs to cover each (feasible) graph shape, one can evaluate the *semantic diversity of a test suite* by calculating how many distinct graph shapes are actually covered.

For example, Figure 3c showcases four models with three equivalence classes (non-isomorphic shapes with a neighbourhood range of 1) for 3-vessel scenarios from an OS viewpoint. We encode the neighbourhoods of OS following the work of [66] to distinguish three shapes (where Shape 3 contains two models with identical neighbourhood shapes). In this paper, logical scenarios are abstracted to functional scenarios to evaluate the semantic diversity of a test suite by graph shapes.

4 Logical Scenarios and Initial Scenes

Next, we formally define logical scenarios represented as *constraint satisfaction problems* (CSP) over numerical (geometric) constraints while initial scenes become solutions of such CSP problems.

4.1 Logical Scenario Models and Initial Scenes

Formally, a *logical scenario model* is a tuple $N := \langle \mathcal{S}_N, C_N, m_N, B_N \rangle$:

- $\mathcal{S}_N$ is a finite set of ships where each $\hat{s}_i \in \mathcal{S}_N$ is an oriented point (see below);
- C_N is a set of binary numerical (geometric) constraints $c_i(\hat{s}_i, \hat{s}_j)$ over ships $\hat{s}_i, \hat{s}_j \in \mathcal{S}_N$;
- m_N is a sea map that restricts the range of position variables for ships in $\mathcal{S}_N$ and
- B_N contains valid ship attribute boundaries (length, cruising speed) as a finite set of tuples $\langle \text{lMin}_i, \text{lMax}_i, \text{spMin}_i, \text{spMax}_i \rangle \in \mathbb{R}^4$.

Given a map m_N and attribute boundaries B_N , we approximate safety domains of ships $\hat{s}_i \in \mathcal{S}_N$ as *oriented points* with a fixed radius represented by a tuple $\hat{s}_i := \langle x, y, h, l, \text{sp} \rangle$, where:

- $x \in \mathbb{R}$ and $y \in \mathbb{R}$ are variables representing the centre point of $\hat{s}_i$.
- $h \in \mathbb{R}$ is a variable representing the heading angle of $\hat{s}_i$ in radians.
- $l \in \mathbb{R}$ is a variable capturing the length of $\hat{s}_i$ in meters.
- $\text{sp} \in \mathbb{R}$ is a variable for the cruising speed of $\hat{s}_i$ given in m/s.

An *initial scene* is a solution of a CSP problem N that defines a logical scenario. An initial scene $\text{scene}_N : \mathcal{S}_N \rightarrow \mathbb{R}^5$ provides an assignment of variables to concrete values within the boundaries such that all numerical (geometric) constraints are satisfied:

- Each ship $\hat{s}_i \in \mathcal{S}_N$ is located within the boundaries: $\text{scene}_N(\hat{s}_i).x \in [0, m_N.x]$, $\text{scene}_N(\hat{s}_i).y \in [0, m_N.y]$, $\text{scene}_N(\hat{s}_i).h \in [-\pi, \pi]$, $\text{scene}_N(\hat{s}_i).l \in [B_N.\text{lMin}_i, B_N.\text{lMax}_i]$, $\text{scene}_N(\hat{s}_i).\text{sp} \in [B_N.\text{spMin}_i, B_N.\text{spMax}_i]$, (where $m_N.x$ and $m_N.y$ represent the width and length of the map).
- Each constraint $c_k(\hat{s}_i, \hat{s}_j) \in C_N$ is satisfied by the assignment scene_N in the context of $\hat{s}_i$ and $\hat{s}_j$.

Note that our definitions do not impose any particular technique on how a solution is obtained.

4.2 Numerical Constraints for COLREGS Situations

We define various numerical constraints $\{c_{\text{coll}}, c_{\text{outVis}}, c_{\text{atVis}}, c_{\text{overtake}}, c_{\text{headOn}}, c_{\text{cross}}\}$ for logical scenarios, which are relevant in the context of COLREGS (see Table 4 for a graphical illustration).

Most constraints rely on relative bearing such as $\angle(\mathbf{p}_{ij}, \mathbf{v}_i)$, which is the angular distance between the relative position $\mathbf{p}_{ij}$ and the velocity $\mathbf{v}_i$ vectors calculated with the function $\angle : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$:

$$\mathbf{p}_{ij} := \begin{pmatrix} \hat{s}_i.x - \hat{s}_j.x \\ \hat{s}_i.y - \hat{s}_j.y \end{pmatrix}^T, \quad \mathbf{v}_i := \hat{s}_i.\text{sp} \times \begin{pmatrix} \cos \hat{s}_i.h \\ \sin \hat{s}_i.h \end{pmatrix}^T, \quad \angle(\mathbf{p}_{ij}, \mathbf{v}_i) := \cos^{-1} \left(\frac{\mathbf{p}_{ij} \cdot \mathbf{v}_i}{\|\mathbf{p}_{ij}\| \|\mathbf{v}_i\|} \right).$$

Table 4. Numeric constraint type definitions (for pairs of ships $\langle \hat{s}_i, \hat{s}_j \rangle \in \mathcal{S}_N^2, \hat{s}_i \neq \hat{s}_j$)

$c_{outVis}(\hat{s}_i, \hat{s}_j) := \mathbf{p}_{ij} \in]D_{begin}(\hat{s}_i, \hat{s}_j), \infty[$ $c_{atVis}(\hat{s}_i, \hat{s}_j) := D_{begin}(\hat{s}_i, \hat{s}_j) - \mathbf{p}_{ij} \in]-\epsilon, \epsilon[$ $c_{coll}(\hat{s}_i, \hat{s}_j) := DCPA_{ij} \in]0, D_{min}(\hat{s}_i, \hat{s}_j)[$	$c_{overtake}(\hat{s}_i, \hat{s}_j) :=$ $(1) \angle(\mathbf{p}_{ji}, \mathbf{v}_j) \in]\frac{\angle_{masthead}}{2}, \pi[\wedge$ $(2) \angle(\mathbf{p}_{ij}, \mathbf{v}_i) \in]0, \frac{\angle_{masthead}}{2}[$
$c_{headOn}(\hat{s}_i, \hat{s}_j) :=$ $(1) \angle(\mathbf{p}_{ji}, \mathbf{v}_j) \in]0, \frac{\angle_{bow}}{2}[\wedge$ $(2) \angle(\mathbf{p}_{ij}, \mathbf{v}_i) \in]0, \frac{\angle_{bow}}{2}[$	$c_{cross}(\hat{s}_i, \hat{s}_j) :=$ $(1) \angle(\mathbf{p}_{ji}, \mathbf{R}_{beam} \cdot \mathbf{v}_j) \in]0, \frac{\angle_{side}}{2}[\wedge$ $(2) \angle(\mathbf{p}_{ij}, \mathbf{v}_i) \in]0, \frac{\angle_{masthead}}{2}[$

Each numerical constraint c_k consists of interval predicates in the form of $val \in]lb, ub[$ connected by logical operators ($\neg, \wedge, \vee$).

Visibility distance constraints c_{outVis} and c_{atVis} . COLREGS Rule 22 specifies the minimum visibility range of a ship's signalling lights based on its length and relative bearing. (Recall from Section 2.2 that the COLREGS rules only apply when both vessels are within visible range of each other.) Let the function $D_{vis}(\hat{s}_i, \hat{s}_j)$ return the visibility distance of $\hat{s}_j$ from $\hat{s}_i$ according to Rule 22. The distance of mutual visibility between $\hat{s}_i$ and $\hat{s}_j$ are the minimum of the relative visibilities:

$$D_{begin}(\hat{s}_i, \hat{s}_j) := \min [D_{vis}(\hat{s}_i, \hat{s}_j), D_{vis}(\hat{s}_j, \hat{s}_i)]. \quad (1)$$

(1) $c_{outVis}(\hat{s}_i, \hat{s}_j) := ||\mathbf{p}_{ij}|| \in]D_{begin}(\hat{s}_i, \hat{s}_j), \infty[$ the distance between the ships is more than $D_{begin}(\hat{s}_i, \hat{s}_j)$, the interaction has not been started.

(2) $c_{atVis}(\hat{s}_i, \hat{s}_j) := D_{begin}(\hat{s}_i, \hat{s}_j) - ||\mathbf{p}_{ij}|| \in]-\epsilon, \epsilon[$ the distance between the ships is precisely $D_{begin}(\hat{s}_i, \hat{s}_j)$, denoting the start of an interaction between the ships (with $\epsilon = e^{-10}$ tolerance).

Closest point of approach. For determining spatial-temporal proximity between two vessels at any given moment, the collision avoidance metrics of *Time to the Closest Point of Approach* (TCPA) and *Distance at the Closest Point of Approach* (DCPA) [37, 80] are used:

$$TCPA_{ij} = \frac{\mathbf{p}_{ij} \cdot \mathbf{v}_{ij}}{||\mathbf{v}_{ij}||^2}, \quad DCPA_{ij} = ||\mathbf{p}_{ji} + \mathbf{v}_{ij} \times \max [0, TCPA_{ij}]|| \quad (2)$$

Assuming unchanged heading and speed attributes, DCPA is the smallest distance possible between the i^{th} and j^{th} vessels during their encounter, whereas TCPA is the time to reach the DCPA.

Collision constraint c_{coll} . The collision constraint enforces that $\hat{s}_i$ and $\hat{s}_j$ are on a collision course. (Recall that COLREGS only apply when risk of collision exists.) We handle each vessel as a circular object with an approximated safety radius that is 4 times [35] its length in all situations ($r_i := 4\hat{s}_i.l$). We use the DCPA to express the collision criterion as an inequality constraint. Let the safety

distance between the vessels be $D_{\min}(\hat{s}_i, \hat{s}_j) = \max(r_i, r_j)$. If the DCPA is less than the minimum distance, then $\hat{s}_i$ and $\hat{s}_j$ are on a collision course: $c_{\text{coll}}(\hat{s}_i, \hat{s}_j) = DCPA_{ij} \in]0, D_{\min}(\hat{s}_i, \hat{s}_j)[$

Relative bearing constraints. Constraints related to overtaking, head-on, and crossing relations are created from two conditions, each pertaining to the bearing of one ship to the other. For overtaking (c_{overtake}), the first condition enforces the bearing constraint of $\hat{s}_j$ to $\hat{s}_i$, ensuring that $\mathbf{p}_{ji}$ is within the stern range ($] \frac{\angle_{\text{masthead}}}{2}, \pi [$) of $\mathbf{v}_j$. The second condition ensures that the bearing constraint of $\hat{s}_i$ to $\hat{s}_j$ places $\mathbf{p}_{ij}$ within the $\angle_{\text{masthead}}$ facing range ($]0, \frac{\angle_{\text{masthead}}}{2}[$) of $\mathbf{v}_i$ (see top right in Table 4). Similarly, we formalize the *head-on* (c_{headOn} , see bottom left in Table 4) and *crossing from port* (c_{cross} , see bottom right in Table 4) using the respective bearing ranges introduced in Section 2.2.

4.3 Abstractions of Logical Scenarios Into Functional Scenarios

A functional model of a COLREGS scenario $F := \langle O_F, I_F \rangle \in \mathcal{F}$ can be derived from a logical scenario model $N := \langle S_N, C_N, m_N, B_N \rangle \in \mathcal{N}$ by an abstraction $\text{l2f} : \mathcal{N} \rightarrow \mathcal{F}$. We derive $\text{l2f}(N)$ by

- (1) abstracting each logical ship $\hat{s}_i \in S_N$ to a corresponding abstract object $o_i \in O_F$: $\hat{s}_i \mapsto o_i$,
- (2) adding unary relations $OS(o_i)$ for own ship (if $i = 1$) and $TS(o_i)$ for target ships (if $i > 1$),
- (3) including binary relations $r_k \in \mathcal{R}_{\text{sil}}$ derived when certain numerical constraints $c_k(\hat{s}_i, \hat{s}_j) \in C_N$ hold over logical ships $\langle \hat{s}_i, \hat{s}_j \rangle \in S_N^2$.

Given the abstraction of logical ships to functional objects $\hat{s}_i \mapsto o_i, \hat{s}_j \mapsto o_j \mid \langle \hat{s}_i, \hat{s}_j \rangle \in S_N^2, \langle o_i, o_j \rangle \in O_F^2$, the abstraction of constraints is defined as follows:

- (1) $c_{\text{overtake}}(\hat{s}_i, \hat{s}_j) \wedge c_{\text{atVis}}(\hat{s}_i, \hat{s}_j) \wedge c_{\text{coll}}(\hat{s}_i, \hat{s}_j) \mapsto \text{overtake}(o_i, o_j)$.
- (2) $c_{\text{headOn}}(\hat{s}_i, \hat{s}_j) \wedge c_{\text{atVis}}(\hat{s}_i, \hat{s}_j) \wedge c_{\text{coll}}(\hat{s}_i, \hat{s}_j) \mapsto \text{headOn}(o_i, o_j) \wedge \text{headOn}(o_j, o_i)$.
- (3) $c_{\text{cross}}(\hat{s}_i, \hat{s}_j) \wedge c_{\text{atVis}}(\hat{s}_i, \hat{s}_j) \wedge c_{\text{coll}}(\hat{s}_i, \hat{s}_j) \mapsto \text{cross}(o_i, o_j)$.
- (4) $c_{\text{outVis}}(\hat{s}_i, \hat{s}_j) \vee \neg c_{\text{coll}}(\hat{s}_i, \hat{s}_j) \mapsto \neg(\text{overtake}(o_i, o_j) \vee \text{headOn}(o_i, o_j) \vee \text{cross}(o_i, o_j))$ (no relation).

Informally, the abstraction complies with the COLREGS such that if the two ships are currently at the edge of their visibility distance and are moving on a collision course, there is a COLREGS relationship between o_i and o_j . If they are not on a collision course or they are outside each other's visibility range, no relation exists between o_i and o_j .

5 Automated Generation of Initial Scenes

Next, we aim to generate initial scenes that provide solutions for a logical scenario. For a given number of ships $K = |S_N| \geq 2$, we aim to solve a CSP problem $N_G := \langle S_N, C_N, m_N, B_N \rangle \in \mathcal{N}$:

$$C_N := \left\{ \langle \hat{s}_i, \hat{s}_j \rangle \in S_N^2 : \begin{cases} c_{\text{outVis}}(\hat{s}_i, \hat{s}_j) \vee \neg c_{\text{coll}}(\hat{s}_i, \hat{s}_j) & \text{if } i \neq 1, j \neq 1 \\ c_{\text{atVis}}(\hat{s}_i, \hat{s}_j) \wedge c_{\text{coll}}(\hat{s}_i, \hat{s}_j) & \text{if } i = 1, j \neq 1 \end{cases} \right\}.$$

Informally, we require that the own ship (**OS**, $i = 1$) is on a collision course with each target ship (**TS**, $i > 1$), and all ships are at the boundary of their mutual visibility ranges. Additionally, no pairs of target ships (**TS**) are on a collision course within each other's visibility range.

To derive solutions of this CSP problem (logical scenario), we adapt two conceptually different approaches, **SB** and **RS**, which are well-known for scenario generation in the automotive domain:

- **SB** uses a search-based approach (see Section 5.1) to derive initial scenes by solving the numerical CSP problem as an optimization problem. This approach is adapted from modern search-based test generation approaches [1, 28, 56] used in autonomous driving.
- **RS** uses *Scenic3* [27, 75], a probabilistic programming language that represents environment models across various domains using high-level constraints. *Scenic3* uses rejection sampling to synthesize concrete scenarios from these high-level descriptions. This involves sampling

from geometric regions (REGION), intervals (RANGE), and then rejecting inconsistent samples based on hard constraints (REQUIRE). **RS** is discussed in depth in Section 5.2.

5.1 Search-Based Scene Generation (SB)

Genetic algorithms (GA) [52, 64] evolve populations of solutions over generations by applying mutation and crossover operations to explore large, complex search spaces even for with highly non-linear problems. We map the numerical CSP of a logical scenario to an *optimization problem* [32] amenable to genetic solvers where a fitness function is minimized (similarly as in [3, 54]).

5.1.1 Solving Logical Scenario Models as an Optimization Problem. We assign a non-negative fitness value for each *candidate solution*, where a fitness value of 0 represents a consistent solution (where all constraints are satisfied). A (multi-objective) *optimization problem* is defined as a tuple $OP = \langle \hat{x}, \hat{l}, \hat{u}, \{f_1(\hat{x}), \dots, f_M(\hat{x})\} \rangle \in \mathcal{OP}$:

- $\hat{x}$ is a vector of *decision variables* (i.e., variables subject to optimization) representing a solution.
- $\hat{l}$ and $\hat{u}$ are vectors of lower and upper bounds for $\hat{x}$.
- $\{f_1(\hat{x}), \dots, f_M(\hat{x})\}$ is a set of objective functions over the variables aimed to be minimized.

Next, we discuss how to derive an optimization problem from a logical scenario model N by a mapping $l2o : N \rightarrow \mathcal{OP}$ to serve as input to any genetic solver (search-based algorithm).

5.1.2 Defining Decision Variables. In GAs, decision variables represent a candidate solution, i.e. one individual in the population. To solve a numerical problem $N_G := \langle \mathcal{S}_N, C_N, m_N, B_N \rangle$, for each $\hat{s} \in \mathcal{S}_N$, we have to find concrete values for $(\hat{s}.x, \hat{s}.y, \hat{s}.h, \hat{s}.l, \hat{s}.sp)$ that satisfy all constraints defined in C_N . Given a number of ships $K = |\mathcal{S}_N|$ and using $\parallel$ to denote concatenation,

- $\hat{x} := \hat{x}^1 \parallel \dots \parallel \hat{x}^K$, where $\hat{x}$ is concatenated from decision variables $\hat{x}^i = \hat{s}_i$.
- $\hat{l} := \hat{l}^1 \parallel \dots \parallel \hat{l}^K$, where each $\hat{l}^i \mid i = 1..K$ is a vector of lower bounds ($\hat{l}^i \leq \hat{x}^i$) constructed as $\hat{l}^i = \langle 0, 0, -\pi, B_N.lMin_i, B_N.spMin_i \rangle$.
- $\hat{u} := \hat{u}^1 \parallel \dots \parallel \hat{u}^K$, where each $\hat{u}^i \mid i = 1..K$ is a vector of upper bounds ($\hat{x}^i \leq \hat{u}^i$) constructed as $\hat{u}^i = \langle m_N.x, m_N.y, \pi, B_N.lMax_i, B_N.spMax_i \rangle$.

5.1.3 Defining the Objective Function. **SB** solves the logical constraint satisfaction problem as an unconstrained optimization problem using GA. To realize this, we define an exterior penalty function [16, 64] that penalizes the deviation of solutions from the given interval bounds.

Penalty function. All $c_k \in C_N$ consists of interval predicates *ipred* in the form of $val \in]lb, ub[$ and logical operators ($\neg, \wedge, \vee$). (3) shows the penalty function defined for non-negated and negated predicates. If two predicates are in an and relation ($\wedge$), we sum up the penalties of two predicates as in (4). If two interval predicates are in an or relation ($\vee$), we aggregate the penalties of the two predicates by taking the minimum penalty as in (5).

$$penalty^+(val, lb, ub) := \begin{cases} lb - val & \text{if } val < lb \\ val - ub & \text{if } val > ub \\ 0 & \text{otherwise.} \end{cases} \quad penalty^-(val, lb, ub) := \min [penalty^+(val, -\infty, lb), penalty^+(val, ub, \infty)]. \quad (3)$$

$$ipred_1 \wedge ipred_2 \mapsto penalty(val_1, lb_1, ub_1) + penalty(val_2, lb_2, ub_2). \quad (4)$$

$$ipred_1 \vee ipred_2 \mapsto \min [penalty(val_1, lb_1, ub_1), penalty(val_2, lb_2, ub_2)]. \quad (5)$$

Using $penalty(val, lb, ub)$ and the penalty aggregation operators over them, we define a function $\Pi : C_N \rightarrow \mathbb{R}_{\geq 0}$, where for all $c \in C_N$, $\Pi(c)$ the overall penalty is derived from the specific constraint. Intuitively, $\Pi(c)$ returns zero if the state of the system satisfies constraint c . Otherwise, it returns

a positive number that is proportional to *how far the system is from satisfying the constraint*. For example, let us consider the penalty value $\Pi(c(\hat{s}_i, \hat{s}_j))$ where $c := \neg c_{coll} \vee c_{outVis}$ which is applied to all pairs of TS vessels. The penalty derivation looks like as follows:

$$\begin{aligned} \Pi(c(\hat{s}_i, \hat{s}_j)) &= \Pi(DCPA_{ij} \notin]0, D_{min}(\hat{s}_i, \hat{s}_j)[\vee ||\mathbf{p}_{ij}|| \in]D_{begin}(\hat{s}_i, \hat{s}_j), \infty[) \\ &= \min [penalty^-(DCPA_{ij}, 0, D_{min}(\hat{s}_i, \hat{s}_j)), penalty^+(||\mathbf{p}_{ij}||, D_{begin}(\hat{s}_i, \hat{s}_j), \infty)]. \end{aligned}$$

Objective function. Many GA algorithms are designed to handle a limited number of objective functions. The fitness values are combined into one or more aggregate objective functions to manage many penalties from various penalty functions. Each aggregation strategy divides the numerical constraints C_N into distinct subsets $C_i \subseteq C_N$, each associated with a specific objective function. In this paper, we use *vessel aggregation* as it is proven to be effective for similar problems in the automotive domain [3]. Vessel aggregation creates subsets for constraints associated with the same vessel, and the penalties in a subset are aggregated together. Formally, given a subset C_i of numerical constraints, an objective function $f_i \in \{f_1, \dots, f_M\}$ is defined as: $f_i = (\sum_{c \in C_i} \Pi(c))$.

5.1.4 Algorithm Selection and Hyper-Parameter Tuning. The choice of aggregation strategy influences the selection of search algorithms. In our multi-vessel setup, aggregation transforms the problem into a multi-objective optimization task. To solve this, we use the NSGA-II algorithm [21], which has proven effective in related automotive studies [3, 74].

Our implementation uses the PyMoo Python library [11] with default operators. We conducted hyper-parameter tuning through a systematic parameter grid search following best practices [25]. The optimized hyper-parameter configuration for our numerical problems of varying complexity is as follows: a *population size* of 10, with *Simulated Binary Crossover* [22] as crossover operator, a *crossover probability* of 1.0, and a *crossover distribution index* η of 20. The *mutation operator* is *Polynomial Mutation* [22], with a *mutation probability* of 0.8 and *mutation distribution index* η of 15. The *selection operator* uses *Binary Tournament with Fitness Domination*.

5.2 Rejection Sampling for Scene Generation (RS)

Our rejection sampling technique **RS** samples variables uniformly from predefined regions (REGION) and intervals (RANGE). A REGION is a 2D shape (e.g., circle, rectangle, cone) that defines a set of possible 2D coordinates for a probabilistic variable. A RANGE is a one-dimensional REGION.

Collision cone region. To be able to sample position and velocity pairs that represent ships on collision courses (satisfying c_{coll}), we use the *Velocity Obstacle method* (VO) [23]. VO is a decentralized motion planning approach that defines feasible velocity regions as geometric cones, enabling quick identification of collision courses. This method is widely adopted for maritime collision avoidance [26, 39, 83]. By definition, if $\mathbf{v}_j$ velocity vector points inside a VOC_{ij} cone region, vessels $\hat{s}_i$ and $\hat{s}_j$ are on a collision course.

5.2.1 Defining the Sampling Region. We provide a high-level overview of how the sampling region is defined for the OS ($\hat{s}_i$) and one TS ($\hat{s}_j$), given a numerical problem $N_G := \langle S_N, C_N, m_N, B_N \rangle \in \mathcal{N}$. The steps are illustrated in Figure 4 along with their visual interpretation.

- (1) First, we sample the TS length $\hat{s}_{j,l}$ from RANGE $[B_N.lMin_j, B_N.lMax_j]$. The position and velocity vector of the OS are sampled from REGION m_N and a circular velocity space REGION with a radius in RANGE $[B_N.spMin_i, B_N.spMax_i]$, respectively.
- (2) We define the possible circular visibility distance REGIONS based on COLREGS Rule 22, with radii 2, 3, 5, or 6 NM, depending on the yet unknown relative bearing between the two ships.
- (3) A TS position is sampled from the union of these REGIONS while satisfying $c_{atVis}(\hat{s}_i, \hat{s}_j)$.

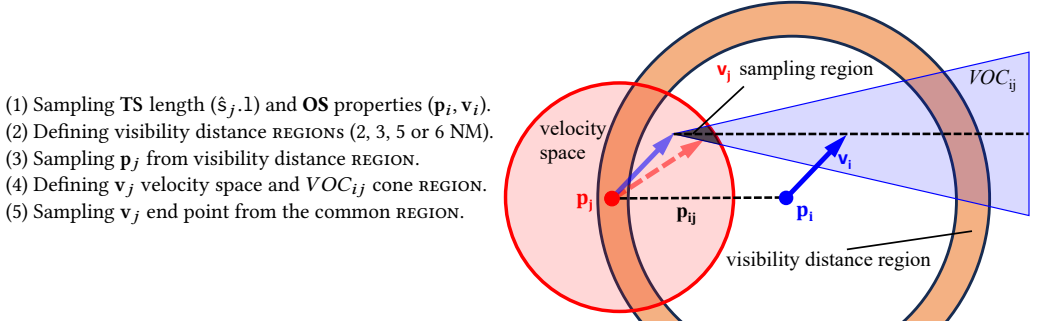


Fig. 4. High-level overview of sampling space definition for the OS and one TS.

Table 5. Used input parameters for $N_G \in \mathcal{N}$ with $i = 1..K$, K denoting the vessel set size $|\mathcal{S}_N|$.

	$B_N.lMin_i$	$B_N.lMax_i$	$B_N.spMin_i$	$B_N.spMax_i$	$m_N.x, m_N.y$
$i = 1$ (OS)	30 m	30 m	0 kn	30 kn (~ 15.43 m/s)	13 NM (~ 24076 m)
$i > 1$ (TS)	10 m	400 m	0 kn	30 kn (~ 15.43 m/s)	

- (4) For the velocity vector of the TS, we define a circular velocity space REGION with a radius in RANGE $[B_N.spMin_j, B_N.spMax_j]$ and a collision cone REGION (VOC_{ij}) of the two vessel.
- (5) Finally, we sample a point from the intersection of the v_j velocity space and VOC_{ij} , which determines the endpoint of v_j , ensuring the satisfaction of $c_{coll}(\hat{s}_i, \hat{s}_j)$.

5.2.2 Rejecting Inconsistent Samples. The visibility distance, consequently the possible positions of the TS and the angle of VOC_{ij} depend on the TS length and the relative bearing between TS and OS. This introduces circular dependencies over probabilistic variables that Scenic3 cannot resolve. To overcome this, we make conservative overestimations, such as using the union of all possible visibility distance REGIONS and the largest possible VOC REGION (when the vessels have maximum length and are at the closest distance of 2 NM).

As a result of overestimated regions, the sampling process may provide inconsistent scenes. To ensure consistency, Scenic3 provides post-sampling filtering in the form of REQUIRE blocks. Since constraints in REQUIRE blocks operate on concrete values rather than probabilistic variables, validity checks are performed as in Section 5.1 by rejecting samples where the penalty exceeds zero.

Rejection sampling (RS) can be extended to multi-vessel scenarios where the no-collision requirement between the TSs is also enforced with REQUIRE blocks.

6 Evaluation

We conducted experiments to address the following research questions in open sea encounters:

- **RQ1:** How relevant are synthetic and real-world test scenarios for COLREGS-compliance?
- **RQ2:** How diverse and complete are the test scenes generated by synthetic approaches?
- **RQ3:** How scalable and fast are synthetic scene generation approaches?

6.1 General Configuration and Measurement Setup

We refer to the number of vessels in a scenario as the *problem size* (K), where $K = |\mathcal{S}_N|$. The input parameters for $N_G \in \mathcal{N}$ are detailed in Table 5, reflecting characteristics of small to large-sized ships. For simplicity, length and speed limits are the same across all TSs. We run the scene generation

Table 6. The ratio of relevant+consistent ($\mathcal{T} \cap C$), and ambiguous+consistent ($\mathcal{A} \cap C$) scenes with respect to successfully generated scenes ($\mathcal{F}$) in the compared datasets.

K	2			3			4			5		6	
Dataset	SB	RS	CO	SB	RS	CO	SB	RS	CO	SB	RS	SB	RS
$\mathcal{F}$ (#)	1000	1000	3222	1000	1000	72	1000	94	4	1000	0	1000	0
$\mathcal{T} \cap C$ (%)	94.4%	94.6%	30.8%	86.8%	89.4%	0%	82.3%	84%	0%	77.3%	0%	68.1%	0%
$\mathcal{A} \cap C$ (%)	0%	0%	0%	39.2%	55.8%	0%	60.4%	80%	0%	66.6%	0%	62%	0%

1000 times with both **SB** and **RS** for each problem size with a timeout of 10 minutes. Finally, we abstract each valid concrete scene using l2f for each batch (i.e. particular approach and problem size) into one of the equivalence classes on the functional level, as described in [Section 4.3](#).

Our measurements are conducted on an Ubuntu 24.04 machine with an Intel Core Processor (Haswell, model: 60, family: 6) and 32 GB of RAM using Python 3.12. The pseudo-random seed is reset to ensure reproducibility, the garbage collector is explicitly called, and two warm-up runs are performed before each measurement to initialize the Python cache, enhancing stability.

6.2 How Relevant Are Synthetic and Real-World Test Scenarios for COLREGS-Compliance?

Motivation. To investigate *relevance* (**D1**), **RQ1** aims to determine to what extent automatically generated (synthetic) and AIS-based (real-world) test scenarios are relevant, i.e., that they represent hazardous COLREGS scenarios to test the COLREGS-compliance of an **OS**.

Metrics. Unfortunately, no generally accepted metrics exist which would measure the relevance of a test suite for assessing COLREGS-compliance. However, a minimum requirement for relevance is that the ships in a test scenario are actually in a COLREGS-scenario (see [Section 2.2](#)). Furthermore, ambiguous scenarios are known to be relevant for system-level testing in the maritime domain [41]. Hence, for each compared approach, we calculate the *ratio of relevant and consistent scenarios* ($\mathcal{T} \cap C$) with respect to the total number of generated scenes. Moreover, we calculate a similar ratio for *ambiguous and consistent* scenes ($\mathcal{A} \cap C$).

Compared approaches. We compare two synthetic scenario generation methods with test scenarios obtained from a real-world dataset.

- The search-based approach **SB** (introduced in [Section 5.1](#)) solves the optimization problem $\text{l2o}(N_G)$, where $N_G \in \mathcal{N}$, using input parameters from [Table 5](#).
- The rejection sampling-based approach **RS** (introduced in [Section 5.2](#)) uses a sampling-based strategy for the same problem.
- The open-source *CommonOcean* benchmark [43] **CO** represents real-world encounter scenarios derived from AIS data collected from various U.S. coastal regions. This benchmark predominantly (97.7%) features two-vessel encounters.

Our techniques **SB** and **RS** are expected to provide fully relevant and consistent scenes faithfully representing COLREGS-scenarios. However, there is no such guarantee for the **CO** benchmark.

Result analysis. [Table 6](#) shows the total number of generated / observed functional scenarios (in the compared datasets), as well as the percentage of relevant+consistent ($\mathcal{T} \cap C$), and ambiguous+consistent ($\mathcal{A} \cap C$) cases with increasing problem size. Since ambiguity is defined only for multi-vessel scenarios, no ambiguous scenarios exist for 2 vessels.

The real-world **CO** dataset has low relevance ratio (30.8%) for $K=2$ and 0% relevance and ambiguous ratio for $K>2$. The low relevance of the dataset prevents its practical use in testing COLREGS-compliance for multi-vessel encounters, thus **CO** is excluded from further investigations below.

In synthetic test suites, a surprising finding is that *not all test scenarios are relevant and consistent*. After manually investigating some of these outliers, we recognized COLREGS-scenarios which are not supposed to exist according to the assumptions on the possible COLREGS-situations. For example, double crossing (when two vessels are mutually crossing from the left) is ruled out by the COLREGS, but it can be judged in case of certain large velocity differences between two vessels.

Technically, **RS** presents slightly more relevant and ambiguous test cases for $K \leq 4$ than **SB**, but it does not provide models for $K > 4$. In general, the ratio of relevant cases decreases, while the ratio of ambiguous cases (within the relevant cases) increases as the problem size increases.

RQ1: Synthetic approaches **SB** and **RS** generate relevant scenes for testing COLREGS compliance (with relevance ratios **68.1%** to **94.6%**). While they also generate ambiguous multi-vessel scenarios, but to a lesser extent (**39%** to **80%**). The real-world **CO** dataset has minimal relevance for multi-ship encounters (**0%** for $K \geq 3$), showing the difficulties of using real AIS data as test scenarios.

6.3 How Diverse and Complete Are the Test Scenes Generated by Synthetic Approaches?

Motivation. Assessing ASV behaviour in semantically diverse scenarios is key for safety assurance (*diversity*, **D2**). Moreover, all relevant and consistent equivalence classes should be covered (*completeness*, **D3**) ideally with similar amount of tests to avoid bias. Thus, **RQ2** explores to what extent the test suites derived by the synthetic approaches cover semantically different scenarios and how the distribution of samples deviates from the even distribution. We measure the coverage for the total number of equivalence classes derived as graph shapes from the functional models.

Compared approaches. We compare the two synthetic scene generation approaches **SB** and **RS**.

Metrics. We use *external shape-based model diversity*, as introduced by [66], along with neighbourhood shapes from Section 3 to distinguish equivalence classes. Our diversity metric is defined as the coverage of relevant and consistent equivalence classes in $\mathcal{T} \cap \mathcal{C}$ by the generated scenes, as well as the coverage for ambiguous and consistent equivalence classes in $\mathcal{A} \cap \mathcal{C}$:

$$\frac{\text{covered shapes in } \mathcal{T} \cap \mathcal{C}}{\text{total shapes in } \mathcal{T} \cap \mathcal{C}}, \quad \frac{\text{covered shapes in } \mathcal{A} \cap \mathcal{C}}{\text{total shapes in } \mathcal{A} \cap \mathcal{C}} \text{ respectively.}$$

Result analysis. In Table 7, we present a summary of the total number of relevant and consistent classes ($|EQ(\mathcal{T} \cap \mathcal{C})|$), and ambiguous and consistent classes ($|EQ(\mathcal{A} \cap \mathcal{C})|$). Moreover, we present the respective equivalence class coverages (Coverage in $\mathcal{T} \cap \mathcal{C}$, and Coverage in $\mathcal{A} \cap \mathcal{C}$).

In Figure 5, each plot illustrates the distribution of equivalence classes within a batch. The x-axis represents the number of samples, while each histogram bar corresponds to an equivalence class. Missing bars indicate that the batch does not cover the respective class.

Both **SB** and **RS** achieve 100% coverage in 2-vessel scenarios. As the number of vessels increases, coverage drops differently for each approach. **SB** shows a nearly linear decline, covering 93.3%, 74.3%, 68.6%, and 49.2% of equivalence classes for 3 to 6 vessels, respectively. In contrast, **RS** declines rapidly, reaching 0% coverage for 5- and 6-vessel scenarios.

Although **SB** is generally more diverse than **RS**, the generation is biased towards certain equivalence classes, and the distribution becomes more skewed as the problem size increases.

Table 7. The total number of equivalence classes of models which are relevant+consistent ($|EQ(\mathcal{T} \cap C)|$), ambiguous+consistent ($|EQ(\mathcal{A} \cap C)|$); Coverage of relevant+consistent ($\mathcal{T} \cap C$), and ambiguous+consistent ($\mathcal{A} \cap C$) equivalence classes by the compared datasets.

K	2		3		4		5		6	
$ EQ(\mathcal{T} \cap C) $ (#)	5		15		35		70		126	
$ EQ(\mathcal{A} \cap C) $ (#)	0		6		21		50		99	
Dataset	SB	RS	SB	RS	SB	RS	SB	RS	SB	RS
Coverage of $\mathcal{T} \cap C$ (%)	100%	100%	93.3%	73.3%	74.2%	31.4%	68.6%	0%	49.2%	0%
Coverage of $\mathcal{A} \cap C$ (%)	0%	0%	100%	100%	85.7%	42.9%	74%	0%	53.5%	0%

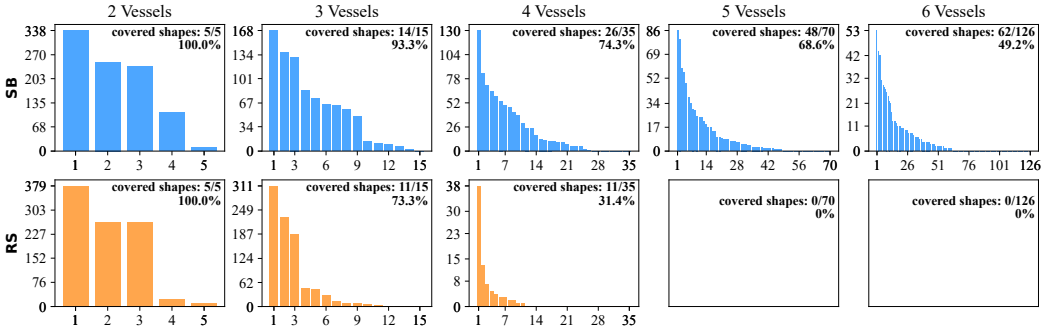


Fig. 5. The distributions of derived scenes across all relevant+consistent equivalence classes for **SB** and **RS**.

When comparing the distributions using the *Chi-Square Test* [30, 58], all the p-values indicate sufficient statistical significance ($p < 0.05$). However, according to *Kullback-Leibler Divergence* [45], for the 2-vessel problem, the distributions of **SB** and **RS** have low divergence, indicating that they are similarly diverse for small problems.

The results indicate that neither of the scene generation approaches can provide sufficient diversity and completeness over the relevant and consistent equivalence classes.

RQ2: **SB** achieves higher coverage than **RS** but coverage decreases with the increasing number of vessels. The decline in coverage and completeness highlights the difficulties of ensuring diverse test case generation as complexity grows.

6.4 How Scalable and Fast Are Synthetic Scene Generation Approaches?

Motivation. Comprehensive testing of ASV behaviour in multi-vessel encounters requires generating a large number of test scenarios for an increasing number of vessels (*scalability*, **D4**) within a limited amount of time (*speed*, **D5**). Since, neither genetic algorithms (used by **SB**) nor rejection sampling (used by **RS**) guarantee that a solution is found, we also assess whether success rates of test generation are adequate for practical use.

Compared approaches. We compare the two synthetic test generation approaches **SB** and **RS**.

Metrics. We evaluate the runtime and success rate for **SB** and **RS** based on scene generation for an increasing numbers of vessels (problem sizes). In particular, we define:

- *Runtime* as the time taken to generate a scenario averaged over successful runs.

Table 8. Success rate, median and average runtime compared to the average time per scene for successful scene generation with the **SB** and **RS** approaches and increasing problem sizes.

K Approach	2		3		4		5		6	
	SB	RS	SB	RS	SB	RS	SB	RS	SB	RS
Success rate (%)	100	100	100	100	100	9.4	100	0	100	0
Median runtime (s)	0.1	0.1	0.33	11.3	0.77	600	2.29	600	7.58	600
Average runtime (s)	0.23	0.15	1.02	16.1	2.2	572	4.9	600	22.6	600
Average time/scene (s)	0.23	0.15	1.02	16.1	2.2	5446.8	4.9	∞	22.6	∞

- *Success rate* as the proportion of runs that produce a valid scenario within a 10-minute timeout. A scene generation is deemed unsuccessful if no solution is found within this limit.
- *Average time/scene* = $\frac{\text{average runtime}}{\text{success rate}}$, estimating the time to successfully synthesize one scene.

Result analysis. Table 8 shows the success rate, as well as the median and average runtime compared to the average time for successful scene generation with the **SB** and **RS** approaches across increasing problem sizes.

SB achieves a **100%** success rate for all measurements, demonstrating its robustness regardless of problem complexity. In contrast, Scenic performs well for smaller cases but its success rate declines as K increases. While it achieves a 100% success rate for $K = 2$ and $K = 3$, its success rate drops significantly to **9.4%** for $K = 4$, and it fails completely (0% success) for $K > 4$.

The median and average runtime of **SB** increases rapidly with the problem size, but it is still manageable with an average of 22.6 seconds for the 6-vessel scenarios. While **RS** outperforms **SB** for $K = 2$, its performance drops drastically, converging to the 10-minute limit for $K = 4$.

After running statistical analysis on the runtimes, we conclude that the results are statistically significant based on the *Mann-Whitney U-test* [48] ($p < 0.05$) with high effect size based on *Cohen's D* [17, 30] ($|e| > 0.8$). This excludes the $K = 2$ case where ($p = 0.072$) and the effect size is small ($e = 0.2$), indicating no evidence that the performance of **SB** and **RS** differs.

The results imply that **SB** is generally faster and more robust for large problems, while **RS** fails to solve logical scenarios when the constraints contain circular dependencies and involve more than 3 actors. These results are aligned with the observations in the automotive domain in [3].

RQ3: **SB** is robust in terms of success rate (100%), and its runtime is acceptable even with an increasing number of actors for open sea encounters. In contrast, **RS** struggles with solving more complex problems ($K > 3$) and it is generally slower than **SB**.

7 Threats to Validity and Limitations

Internal validity. This work aims to compare the direct adaptations of two scenario generation approaches popular in the automotive domain to evaluate their effectiveness in the maritime domain. We have striven to adapt the Scenic approach **RS** as closely as possible to the maritime domain. When implementing **SB**, we applied various best practices, such as choosing a widely used algorithm and library for search-based test generation (NSGA-II), using hyperparameter tuning, and applying proven design decisions from existing state-of-the-art work. Nevertheless, we cannot fully guarantee that we provide the best possible implementation, hence it is a potential threat to validity. Furthermore, the success rate of **RS** would likely increase with an increased timeout.

External validity. Due to the informal nature of the COLREGS, it is not possible to formulate a canonical model of COLREGS-compliant behaviour, which constitutes a threat to external validity (i.e., a different formalization of COLREGS may yield different empirical findings). In particular, we

do not consider the special COLREGS regulations for vessels with limited manoeuvrability, such as sailing vessels (Rule 18), which is left to future work. Moreover, we exclude dynamic properties from our models, such as the emergence or disappearance of new COLREGS relations due to evasive actions taken by actors. To handle such cases, our models need to incorporate temporal behaviour in the future. In terms of logical model constraints, we use meaningful domain-specific parameters aligned with standards [19, 77], but have chosen a simpler circular safety domain [35, 39, 42, 84] in favour of one of the more advanced elliptical domains proposed in the literature [15, 18, 60, 69, 77].

Despite potential limitations of our COLREGS model, we believe that it accurately captures the two most important differences between the automotive and maritime domains, namely, the unconstrained scenario geometry, and the underspecified nature of Rules 12–18.

We investigate test scenarios derived from historical AIS data in the context of the **CO** benchmark, and obtain negative results. While findings might be different in the context of other AIS data, it is highly unlikely that higher relevance would be observed when testing multi-vessel encounters since such encounters are rare in real life, though they present particularly challenging cornercases. Moreover, deriving actual test scenarios from real-world AIS data is a complex challenge.

8 Discussion and Lessons Learned

After the evaluation, we aim to summarize the practical takeaways of our investigations related to the various quality requirements set up for scenario generators in a maritime context.

Scenarios derived from real-world AIS data are insufficient for testing multi-ship encounters. The lack of a canonical formalization of COLREGS discussed in Section 7 poses a challenge also when extracting test scenarios from historical traffic data. Varying notions of what constitutes a relevant scenario are used across different studies [42, 76, 85], which complicates a meaningful comparison of datasets. As a practical example of this problem, the authors of the **CO** dataset state in [43] that scenarios were selected on the basis of demonstrating “close encounters” between vessels. However, our study finds that the majority of scenarios in **CO** do not actually meet our definition of relevant scenarios, as the vessels involved were not in situations regulated by COLREGS (e.g., they were not on potential collision courses). Furthermore, multi-ship encounters occur very rarely in test scenarios derived from AIS data [43, 76, 85]. Thus, such encounters represent rare, but potentially safety-critical edge cases which are unlikely to be sufficiently tested by scenarios derived from existing traffic. All of this clearly highlights the *need for systematic synthetic scenario generators for the maritime domain*, where the relevance of test cases can be ensured by construction.

Adapting automotive scenario generators may provide sufficient scalability and speed in maritime. As a positive finding of our paper, adapting existing scenario generation techniques that proved to be effective in the automotive domain is *technically feasible* both for search-based and rejection sampling based approaches. Moreover, *search-based scenario generation provided sufficient scalability and speed* for deriving test scenarios for open sea encounters with increasing number of vessels for problems of practical size (3–6 vessels). On the other hand, while rejection sampling was proven highly beneficial in the automotive domain, it was not effective in generating maritime test scenarios, most probably, due to the higher degrees of freedom in vessel placement.

Adapting existing techniques lacks diversity and completeness in a maritime setting. As a negative finding and major limitation of existing approaches, both **SB** and **RS** failed to cover many equivalence classes, particularly for larger problems with more vessels. This highlights significant quality issues in directly adapting scenario generation techniques from automotive to maritime domains. Moreover, it clearly shows that further research is necessitated to provide mechanisms that enforce diversity and ensure completeness (on a functional level) during test scenario generation.

Potential gaps in COLREGS-assumptions. As a highly unexpected finding (reported in Table 6) synthetic test generation techniques repeatedly derived scenarios where the functional scenario model was inconsistent. This is particularly surprising due to the following reasons:

- By construction, we ensure that the initial scenes derived by the algorithms presented in Section 5 enforce COLREGS-situations between the own ship and the target ships (hence we ensure relevance of the test cases).
- The consistency constraints of our functional scenario models (Table 2) were validated manually by maritime experts and automatically by graph generation techniques, hence we had strong confidence in that they faithfully represent the key assumptions and restrictions of COLREGS on a high level of abstraction.
- Assuming that the algorithms are correct and our formalization is faithful, then the detected discrepancies may indicate potentially invalid assumptions in some COLREGS-situations.

After further investigations, we have found several cases, which are ruled out by the COLREGS (e.g. double crossing from port, when two vessels would be considered to be mutually crossing from the left). Nevertheless, there may be very rare situations when they can be potentially observed, e.g. when there is a large (but not unrealistic) difference between the velocity of two vessels. Investigating the actual existence of such cases in a practical setting is subject to future work.

9 Related Work

Formalization of COLREGS. A recent study of COLREGS [41] presents a functional metamodel for representing COLREGS scenarios and uses automated model generation to support ASV test case synthesis. We adopt their formalization to explore functional diversity and extend it to accommodate scenarios involving an arbitrary number of vessels.

For ASV navigation, the authors in [35] present a framework for quantitative analysis of COLREGS compliance. Similarly, in [44] and [42], the authors provide a temporal-logic formalization of COLREGS and use it to verify compliance from recorded ship trajectories. However, their COLREGS formalizations are unsuitable for test-scenario synthesis since they do not support deriving concrete scenarios. Moreover, [42] does not take visibility range (Rule 11) into account.

The abstraction of concrete scenes is a means to establish COLREGS situation awareness. In [20, 47], the authors leverage COLREGS formalization to reactively plan the ASV's next manoeuvre based on the high-level situation.

Scenario generation for ASVs. Existing work either relies on manually designed test scenarios [10, 47, 53, 68, 79] or the test scenarios are derived from historical, real-world AIS data, as explored in studies such as [5, 40, 43, 46, 57, 76, 85]. The open-source CommonOcean benchmark [43] adopts a time-frame-based extraction method, analyzing vessel movements within specified temporal and spatial windows. In contrast, other research, including [5, 76, 85], employs proximity-based metrics such as relative distance, TCPA, and DCPA with varying thresholds to identify scenarios. Unlike these approaches, our logical scenario models are directly based on the COLREGS.

Some previous work also involves synthetic scenario generation. In [67], a state machine-based scenario synthesis is proposed. However, the authors do not account for COLREGS. In [61], the authors propose an online scenario generation and selection algorithm with a discriminative neural network to assess ASV behaviour in high-risk COLREGS scenarios. However, their approach gives no assurance on the diversity of the generated scenarios. Two papers that attempt to address this limitation are [12] and [70]. In the former, traffic scenarios are generated using Sobol sequences. Clustering algorithms then group scenarios based on assigned risk vectors. In [12], equivalence classes are not defined based on COLREGS but on a multi-dimensional risk vector, making the interpretation of an equivalence class unclear. In [70], the authors generate ASV test scenarios by

formulating requirements in Signal Temporal Logic, complemented with a Gaussian Process model to guide the test case selection automatically. Similarly to our experiment, in [33], the authors use "COLREGS monitors" for abstraction and logical scenarios to derive concrete scenes. However, they use combinatorial test generation, which is unlikely to scale for larger vessel counts.

Scenario generation in the automotive domain. On the logical level, traffic synthesis for ASVs shares some similarities with the corresponding problem for autonomous road vehicles (actors, orientations) [4, 51, 71]. Search-based algorithms are one of the most common solutions for generating concrete traffic scenarios in the automotive field [1, 6–8, 34, 55, 63]. The approach presented in [3] use partial modelling and metaheuristic search to derive concrete scenes from abstract traffic scenarios. While we adapt their three abstraction layers from automotive to maritime domain, our paper does not use functional scenarios to guide for scene generation, but rather to evaluate coverage. The authors of [13] propose a search-based approach to generate avoidable collision scenarios for autonomous driving systems.

Sampling-based approaches are also popular in synthesizing traffic scenarios [54]. The Scenic3 framework [27, 75] offers a rejection sampling based generation of 2D or 3D scenes and dynamic scenarios while accepting any map as an input parameter. Scenic3 primarily supports the automotive domain [81], but other domains (maritime, airborne) can also be implemented.

10 Conclusions

In this experience report, we investigate to what extent existing test scenario generation approaches developed in an automotive context can be adapted for the maritime domain. For that purpose, we define logical scenarios in a maritime context, and adapt two conceptually different approaches to derive initial scenes from logical scenarios: (a) a search-based approach **SB** using genetic algorithms, and (b) a rejection sampling based approach **RS** based on Scenic3. We evaluate the two synthetic scenario generation approaches as well as test scenarios obtained from real-world AIS data from the CommonOcean benchmark with respect to high-level requirements such as relevance, diversity, completeness, scalability and speed.

Our findings indicate that the adaptation of test generation techniques from the automotive domain is technically feasible and search-based techniques provide sufficient speed and scalability, but their performance is limited w.r.t. diversity and completeness. Furthermore, test scenarios derived from real-world AIS data are insufficient for assuring COLREGS-compliance in multi-ship encounters. Finally, some of the derived test scenes also provide evidence that certain assumptions made in COLREGS may be violated in some edge cases.

In future work, we aim to extend the functional and logical scenarios to incorporate multiple ship classes (including vessels with limited manoeuvrability) and environmental factors (e.g. wind speed, fog). Moreover, our experience paper highlights a clear need to improve the diversity and completeness of existing test scenario generation techniques. Finally, similar investigation would be meaningful in the context of complex, traffic-heavy environments, such as ports or coastal regions.

11 Data Availability

We released all the experimental data and source code in the project repository for replication, future research, and practical use: <https://github.com/PELAB-LiU/ASVTestSceneGeneration>.

Acknowledgments

This work relates to the Department of Navy Award N62909-24-1-2006 issued by the Office of Naval Research. The third author has also been partially supported by the Wallenberg AI, Autonomous Systems, and Software Program (WASP), which is funded by the Knut and Alice Wallenberg

Foundation, and Vetenskapsrådet (VR). The first author has been supported by the Linköping University Graduate School in Computer Science (CUGS) and is also affiliated with WASP.

References

- [1] Raja Ben Abdesslem, Annibale Panichella, Shiva Nejati, Lionel C. Briand, and Thomas Stifter. 2018. Testing autonomous cars for feature interaction failures using many-objective search. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering (ASE '18)*. ACM, 143–154. doi:10.1145/3238147.3238192
- [2] Arnold Akkermann and Axel Hahn. 2020. Comparing Simulation with Physical Verification and Validation in a Maritime Test Field. *International Journal of Systems Engineering* 4, 2 (2020), 18. doi:10.11648/j.ijse.20200402.12
- [3] Aren A. Babikian, Oszkár Semeráth, and Dániel Varró. 2024. Concretization of Abstract Traffic Scene Specifications Using Metaheuristic Search. *IEEE Transactions on Software Engineering* 50, 1 (2024), 48–68. doi:10.1109/tse.2023.3331254
- [4] Gerrit Bagschik, Till Menzel, and Markus Maurer. 2018. Ontology based Scene Creation for the Development of Automated Vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 1813–1820. doi:10.1109/ivs.2018.8500632
- [5] Azzeddine Bakdi, Ingrid Kristine Glad, and Erik Vanem. 2021. Testbed Scenario Design Exploiting Traffic Big Data for Autonomous Ship Trials Under Multiple Conflicts With Collision/Grounding Risks and Spatio-Temporal Dependencies. *IEEE Transactions on Intelligent Transportation Systems* 22, 12 (2021), 7914–7930. doi:10.1109/tits.2021.3095547
- [6] Raja Ben Abdesslem, Shiva Nejati, Lionel C. Briand, and Thomas Stifter. 2016. Testing advanced driver assistance systems using multi-objective search and neural networks. In *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering (Singapore, Singapore) (ASE '16)*. Association for Computing Machinery, New York, NY, USA, 63–74.
- [7] Raja Ben Abdesslem, Shiva Nejati, Lionel C. Briand, and Thomas Stifter. 2016. Testing advanced driver assistance systems using multi-objective search and neural networks. In *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering (ASE'16)*. ACM, 63–74.
- [8] Raja Ben Abdesslem, Shiva Nejati, Lionel C. Briand, and Thomas Stifter. 2018. Testing Vision-Based Control Systems Using Learnable Evolutionary Algorithms. In *40th International Conference on Software Engineering (ICSE '18)*. ACM, 1016–1026. doi:10.1145/3180155.3180160
- [9] Gábor Bergmann, Zoltán Ujhelyi, István Ráth, and Dániel Varró. 2011. *A Graph Query Language for EMF Models*. Springer Berlin Heidelberg, 167–182. doi:10.1007/978-3-642-21732-6_12
- [10] Ivan R. Bertaska, Brual Shah, Karl von Ellenrieder, Petr Švec, Wilhelm Klinger, Armando J. Sinisterra, Manhar Dhanak, and Satyandra K. Gupta. 2015. Experimental evaluation of automatically-generated behaviors for USV operations. *Ocean Engineering* 106 (2015), 496–514. doi:10.1016/j.oceaneng.2015.07.002
- [11] J. Blank and K. Deb. 2020. pymoo: Multi-Objective Optimization in Python. *IEEE Access* 8 (2020), 89497–89509.
- [12] Victor Bolbot, Christos Gkerekos, Gerasimos Theotokatos, and Evangelos Boulougouris. 2022. Automatic traffic scenarios generation for autonomous ships collision avoidance system testing. *Ocean Engineering* 254 (2022), 111309. doi:10.1016/j.oceaneng.2022.111309
- [13] Alessandro Calo, Paolo Arcaini, Shaikat Ali, Florian Hauer, and Fuyuki Ishikawa. 2020. Generating Avoidable Collision Scenarios for Testing Autonomous Driving Systems. In *2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST)*. IEEE, 375–386. doi:10.1109/icst46399.2020.00045
- [14] A.N. Cockcroft and J.N.F. Lameijer. 2012. *Steering and sailing rules*. Elsevier, 11–104. doi:10.1016/b978-0-08-097170-4.00002-7
- [15] Alfred Norman Cockcroft and Jan Nanne Frits Lameijer. 2012. *Guide to the collision avoidance rules*. Elsevier. doi:10.1016/c2010-0-68322-2
- [16] Carlos A Coello Coello. 2002. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering* 191, 11–12 (Jan. 2002), 1245–1287. doi:10.1016/s0045-7825(01)00323-1
- [17] Jacob Cohen. 2013. *Statistical Power Analysis for the Behavioral Sciences*. Routledge. doi:10.4324/9780203771587
- [18] TG Coldwell. 1983. Marine traffic behaviour in restricted waters. *The Journal of Navigation* 36, 3 (1983), 430–444.
- [19] UCG Commandant. 1999. International regulations for prevention of collisions at sea, 1972 (72 COLREGS). US Department of Transportation, US Coast Guard, Commandant Instruction M 16672 (1999).
- [20] James Connor Meyers, Thomas Sayre-McCord, Zhiyong Zhang, and Hanumant Singh. 2024. Toward a COLREGs Compliant Autonomous Surface Vessel in a Constrained Channel. *IEEE Transactions on Field Robotics* 1 (2024), 537–547. doi:10.1109/TFR.2024.3457273
- [21] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197. doi:10.1109/4235.996017
- [22] Kalyanmoy Deb, Karthik Sindhya, and Tatsuya Okabe. 2007. Self-adaptive simulated binary crossover for real-parameter optimization. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO '07)*. ACM,

- 1187–1194. doi:10.1145/1276958.1277190
- [23] T. Degré and X. Lefèvre. 1981. A Collision Avoidance System. *Journal of Navigation* 34, 2 (1981), 294–302. doi:10.1017/s0373463300021408
- [24] Jun Ding, Yinxuan Zhou, Xia Huang, Kun Song, Shiqing Lu, and Lusheng Wang. 2023. An improved RRT* algorithm for robot path planning based on path expansion heuristic sampling. *Journal of Computational Science* 67 (2023), 101937. doi:10.1016/j.jocs.2022.101937
- [25] A. E. Eiben and S. K. Smit. 2011. *Evolutionary Algorithm Parameters and Methods to Tune Them*. Springer Berlin Heidelberg, 15–36. doi:10.1007/978-3-642-21434-9_2
- [26] Paolo Fiorini and Zvi Shiller. 1998. Motion Planning in Dynamic Environments Using Velocity Obstacles. *The International Journal of Robotics Research* 17, 7 (1998), 760–772. doi:10.1177/027836499801700706
- [27] Daniel J. Fremont, Tommaso Dreossi, Shromona Ghosh, Xiangyu Yue, Alberto L. Sangiovanni-Vincentelli, and Sanjit A. Seshia. 2019. Scenic: a language for scenario specification and scene generation. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '19)*. ACM, 63–78. doi:10.1145/3314221.3314633
- [28] Christoph Gladisch, Thomas Heinz, Christian Heinzemann, Jens Oehlerking, Anne von Vietinghoff, and Tim Pfitzer. 2019. Experience Paper: Search-Based Testing in Automated Driving Control Applications. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 26–37. doi:10.1109/ase.2019.00013
- [29] Denis Gopan, Frank DiMaio, Nurit Dor, Thomas Reps, and Mooly Sagiv. 2004. *Numeric Domains with Summarized Dimensions*. Springer Berlin Heidelberg, 512–529. doi:10.1007/978-3-540-24730-2_38
- [30] Robert J. Grissom and John J. Kim. 2014. *Effect Sizes for Research*. Routledge. doi:10.4324/9781410612915
- [31] Baptiste Gueuziec, Jean-Pierre Gallois, and Frédéric Boulanger. 2024. Qualitative reasoning and cyber-physical systems: abstraction, modeling, and optimized simulation. *Innovations in Systems and Software Engineering* 20, 4 (2024), 511–529. doi:10.1007/s11334-024-00567-0
- [32] Nyoman Gunantara. 2018. A review of multi-objective optimization: Methods and its applications. *Cogent Engineering* 5, 1 (Jan. 2018), 1502242. doi:10.1080/23311916.2018.1502242
- [33] Georg Hake, David Reiher, Jan Mentjes, and Axel Hahn. 2024. Integrating scenario- and contract-based verification for automated vessels. *Journal of Marine Science and Technology* 29, 3 (2024), 620–640. doi:10.1007/s00773-024-01008-0
- [34] Fitash Ul Haq, Donghwan Shin, and Lionel Briand. 2022. Efficient online testing for DNN-enabled systems using surrogate-assisted and many-objective optimization. In *Proceedings of the 44th International Conference on Software Engineering (Pittsburgh, Pennsylvania) (ICSE '22)*. Association for Computing Machinery, New York, NY, USA, 811–822. doi:10.1145/3510003.3510188
- [35] Yixiong He, Yi Jin, Liwen Huang, Yong Xiong, Pengfei Chen, and Junmin Mou. 2017. Quantitative analysis of COLREG rules and seamanship for autonomous collision avoidance at open sea. *Ocean Engineering* 140 (2017), 281–291. doi:10.1016/j.oceaneng.2017.05.029
- [36] Thomas Helmer, Lei Wang, Klaus Kompass, and Ronald Kates. 2015. Safety Performance Assessment of Assisted and Automated Driving by Virtual Experiments: Stochastic Microscopic Traffic Simulation as Knowledge Synthesis. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2019–2023. doi:10.1109/itsc.2015.327
- [37] Helmut Hilgert and Michael Baldauf. 1997. A common risk model for the assessment of encounter situations on board ships. *Deutsche Hydrographische Zeitschrift* 49, 4 (1997), 531–542. doi:10.1007/bf02764347
- [38] Liang Hu, Huosheng Hu, Wasif Naeem, and Zidong Wang. 2022. A review on COLREGs-compliant navigation of autonomous surface vehicles: From traditional to learning-based approaches. *Journal of Automation and Intelligence* 1, 1 (2022), 100003. doi:10.1016/j.jai.2022.100003
- [39] Yamin Huang, P.H.A.J.M. van Gelder, and Yuanqiao Wen. 2018. Velocity obstacle algorithms for collision prevention at sea. *Ocean Engineering* 151 (2018), 308–321. doi:10.1016/j.oceaneng.2018.01.001
- [40] Taewoong Hwang and Ik-Hyun Youn. 2021. Navigation Situation Clustering Model of Human-Operated Ships for Maritime Autonomous Surface Ship Collision Avoidance Tests. *Journal of Marine Science and Engineering* 9, 12 (2021), 1458. doi:10.3390/jmse9121458
- [41] Ulf Kargén and Dániel Varró. 2024. Towards Automated Test Scenario Generation for Assuring COLREGs Compliance of Autonomous Surface Vehicles. In *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems (MODELS '24, Vol. 222)*. ACM, 249–256. doi:10.1145/3640310.3674098
- [42] Hanna Krasowski and Matthias Althoff. 2021. Temporal Logic Formalization of Marine Traffic Rules. In *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 186–192. doi:10.1109/iv48863.2021.9575685
- [43] Hanna Krasowski and Matthias Althoff. 2022. CommonOcean: Composable Benchmarks for Motion Planning on Oceans. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 1676–1682. doi:10.1109/itsc55140.2022.9921925
- [44] Arne Kreutzmann, Diedrich Wolter, Frank Dylla, and Jae Hee Lee. 2013. Towards Safe Navigation by Formalizing Navigation Rules. *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation* 7, 2 (2013), 161–168. doi:10.12716/1001.07.02.01

- [45] S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics* 22, 1 (1951), 79–86. doi:10.1214/aoms/1177729694
- [46] Anton Kullberg, Isaac Skog, and Gustaf Hendeby. 2021. Learning Motion Patterns in AIS Data and Detecting Anomalous Vessel Behavior. In *2021 IEEE 24th International Conference on Information Fusion (FUSION)*. IEEE, 1–8. doi:10.23919/fusion49465.2021.9627027
- [47] Yoshiaki Kuwata, Michael T. Wolf, Dimitri Zarzhitsky, and Terrance L. Huntsberger. 2014. Safe Maritime Autonomous Navigation With COLREGS, Using Velocity Obstacles. *IEEE Journal of Oceanic Engineering* 39, 1 (2014), 110–119. doi:10.1109/JOE.2013.2254214
- [48] H. B. Mann and D. R. Whitney. 1947. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics* 18, 1 (1947), 50–60. doi:10.1214/aoms/1177730491
- [49] Kristóf Marussy, Attila Ficsor, Oszkár Semeráth, and Dániel Varró. 2024. Refinery: Graph Solver as a Service: Refinement-based Generation and Analysis of Consistent Models. In *Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion '24, Vol. 21)*. ACM, 64–68. doi:10.1145/3639478.3640045
- [50] Kristóf Marussy, Oszkár Semeráth, and Dániel Varró. 2022. Automated Generation of Consistent Graph Models With Multiplicity Reasoning. *IEEE Transactions on Software Engineering* 48, 5 (2022), 1610–1629. doi:10.1109/tse.2020.3025732
- [51] Till Menzel, Gerrit Bagschik, Leon Isensee, Andre Schomburg, and Markus Maurer. 2019. From Functional to Logical Scenarios: Detailing a Keyword-Based Scenario Description for Execution in a Simulation Environment. In *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2383–2390. doi:10.1109/ivs.2019.8814099
- [52] Seyedali Mirjalili. 2018. *Genetic Algorithm*. Springer International Publishing, 43–55. doi:10.1007/978-3-319-93025-1_4
- [53] Wasif Naeem, George W. Irwin, and Aoilei Yang. 2012. COLREGs-based collision avoidance strategies for unmanned surface vehicles. *Mechatronics* 22, 6 (2012), 669–678. doi:10.1016/j.mechatronics.2011.09.012
- [54] Matthew O’Kelly, Aman Sinha, Hongseok Namkoong, John Duchi, and Russ Tedrake. 2018. Scalable End-to-End Autonomous Vehicle Testing via Rare-event Simulation. doi:10.48550/ARXIV.1811.00145
- [55] Annibale Panichella, Fitsum Meshesha Kifetew, and Paolo Tonella. 2015. Reformulating Branch Coverage as a Many-Objective Optimization Problem. In *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*. IEEE, 1–10. doi:10.1109/icst.2015.7102604
- [56] Annibale Panichella, Fitsum Meshesha Kifetew, and Paolo Tonella. 2018. Automated Test Case Generation as a Many-Objective Optimisation Problem with Dynamic Selection of the Targets. *IEEE Transactions on Software Engineering* 44, 2 (2018), 122–158. doi:10.1109/tse.2017.2663435
- [57] Myron Papadimitrakis, Marios Stogiannos, Haralambos Sarimveis, and Alex Alexandridis. 2021. Multi-Ship Control and Collision Avoidance Using MPC and RBF-Based Trajectory Predictions. *Sensors* 21, 21 (2021), 6959. doi:10.3390/s21216959
- [58] Karl Pearson. 1900. X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 50, 302 (1900), 157–175. doi:10.1080/14786440009463897
- [59] Tom Arne Pedersen, Jon Arne Glomsrud, Else-Line Ruud, Aleksander Simonsen, Jarle Sandrib, and Bjørn-Olav Holtung Eriksen. 2020. Towards simulation-based verification of autonomous navigation systems. *Safety Science* 129 (2020), 104799. doi:10.1016/j.ssci.2020.104799
- [60] Zbigniew Pietrzykowski and Mirosław Wielgosz. 2021. Effective ship domain – Impact of ship size and speed. *Ocean Engineering* 219 (2021), 108423. doi:10.1016/j.oceaneng.2020.108423
- [61] Ivan Porres, Sepinoud Azimi, and Johan Lilius. 2020. Scenario-based Testing of a Ship Collision Avoidance System. In *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 545–552. doi:10.1109/seaa51224.2020.00090
- [62] Arend Rensink and Dino Distefano. 2006. Abstract Graph Transformation. *Electronic Notes in Theoretical Computer Science* 157, 1 (2006), 39–59. doi:10.1016/j.entcs.2006.01.022
- [63] Vincenzo Riccio and Paolo Tonella. 2020. Model-based exploration of the frontier of behaviours for deep learning system testing. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE ’20)*. ACM, 876–888. doi:10.1145/3368089.3409730
- [64] Jon T. Richardson, Mark R. Palmer, Gunar E. Liepins, and Mike R. Hilliard. 1989. Some Guidelines for Genetic Algorithms with Penalty Functions. In *Proceedings of the 3rd International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 191–197.
- [65] Oszkár Semeráth, Aren A. Babikian, Anqi Li, Kristóf Marussy, and Dániel Varró. 2020. Automated generation of consistent models with structural and attribute constraints. In *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (Virtual Event, Canada) (MODELS ’20)*. Association for Computing Machinery, New York, NY, USA, 187–199. doi:10.1145/3365438.3410962

- [66] Oszkár Semeráth, Rebeka Farkas, Gábor Bergmann, and Dániel Varró. 2019. Diversity of graph models and graph generators in mutation testing. *International Journal on Software Tools for Technology Transfer* 22, 1 (2019), 57–78. doi:10.1007/s10009-019-00530-6
- [67] Hamed Yaghoubi Shahir, Uwe Glässer, Roozbeh Farahbod, Piper Jackson, and Hans Wehn. 2012. Generating test cases for marine safety and security scenarios: a composition framework. *Security Informatics* 1, 1 (2012), 4. doi:10.1186/2190-8532-1-4
- [68] Paul G. Stankiewicz and Galen E. Mullins. 2019. Improving Evaluation Methodology for Autonomous Surface Vessel COLREGS Compliance. In *OCEANS 2019 - Marseille*. IEEE, 1–7. doi:10.1109/oceanse.2019.8867549
- [69] Rafal Szlapczynski and Joanna Szlapczynska. 2017. Review of ship safety domains: Models and applications. *Ocean Engineering* 145 (Nov. 2017), 277–289. doi:10.1016/j.oceaneng.2017.09.020
- [70] Tobias Rye Torben, Jon Arne Glomsrud, Tom Arne Pedersen, Ingrid B Utne, and Asgeir J Sørensen. 2022. Automatic simulation-based testing of autonomous ships using Gaussian processes and temporal logic. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 237, 2 (2022), 293–313. doi:10.1177/1748006x211069277
- [71] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer. 2015. Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems (2015 IEEE 18th International Conference on Intelligent Transportation Systems)*. IEEE, 982–988. doi:10.1109/ITSC.2015.164
- [72] Anete Vagale, Ottar L Osen, Andreas Brandsæter, Marius Tannum, Christian Hovden, and Robin T Bye. 2022. On the use of maritime training simulators with humans in the loop for understanding and evaluating algorithms for autonomous vessels. *Journal of Physics: Conference Series* 2311, 1 (2022), 012026. doi:10.1088/1742-6596/2311/1/012026
- [73] Anete Vagale, Rachid Oucheikh, Robin T. Bye, Ottar L. Osen, and Thor I. Fossen. 2021. Path planning and collision avoidance for autonomous surface vehicles I: a review. *Journal of Marine Science and Technology* 26, 4 (2021), 1292–1306. doi:10.1007/s00773-020-00787-6
- [74] Shanu Verma, Millie Pant, and Vaclav Snasel. 2021. A Comprehensive Review on NSGA-II for Multi-Objective Combinatorial Optimization Problems. *IEEE Access* 9 (2021), 57757–57791. doi:10.1109/access.2021.3070634
- [75] Eric Vin, Shun Kashiwa, Matthew Rhea, Daniel J. Fremont, Edward Kim, Tommaso Dreossi, Shromona Ghosh, Xiangyu Yue, Alberto L. Sangiovanni-Vincentelli, and Sanjit A. Seshia. 2023. *3D Environment Modeling for Falsification and Beyond with Scenic 3.0*. Springer Nature Switzerland, 253–265. doi:10.1007/978-3-031-37706-8_13
- [76] Weiqiang Wang, Liwen Huang, Kezhong Liu, Yang Zhou, Zhitao Yuan, Xuri Xin, and Xiaolie Wu. 2024. Ship encounter scenario generation for collision avoidance algorithm testing based on AIS data. *Ocean Engineering* 291 (2024), 116436. doi:10.1016/j.oceaneng.2023.116436
- [77] Miroslaw Wielgosz and Zbigniew Pietrzykowski. 2022. The ship domain in navigational safety assessment. *PLOS ONE* 17, 4 (2022), e0265681. doi:10.1371/journal.pone.0265681
- [78] Kyle Woerner, Michael R. Benjamin, Michael Novitzky, and John J. Leonard. 2018. Quantifying protocol evaluation for autonomous collision avoidance: Toward establishing COLREGS compliance metrics. *Autonomous Robots* 43, 4 (2018), 967–991. doi:10.1007/s10514-018-9765-y
- [79] Kyle L. Woerner, Michael R. Benjamin, Michael Novitzky, and John J. Leonard. 2016. Collision avoidance road test for COLREGS-constrained autonomous vehicles. In *OCEANS 2016 MTS/IEEE Monterey*. IEEE, 1–6. doi:10.1109/oceans.2016.7761413
- [80] Qingyang Xu and Ning Wang. 2014. A Survey on Ship Collision Risk Evaluation. *PROMET - Traffic&Transportation* 26, 6 (2014), 475–486. doi:10.7307/ptt.v26i6.1386
- [81] Jiawei Zhang, Chejian Xu, and Bo Li. 2024. ChatScene: Knowledge-Enabled Safety-Critical Scenario Generation for Autonomous Vehicles. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 15459–15469. doi:10.1109/cvpr52733.2024.01464
- [82] Songan Zhang, Huei Peng, Ding Zhao, and H. Eric Tseng. 2018. Accelerated Evaluation of Autonomous Vehicles in the Lane Change Scenario Based on Subset Simulation Technique. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 3935–3940. doi:10.1109/itsc.2018.8569800
- [83] Xiaoxue Zhang, Jun Ma, Zilong Cheng, Masayoshi Tomizuka, and Tong Heng Lee. 2022. Velocity Obstacle Based Risk-Bounded Motion Planning for Stochastic Multi-Agent Systems. doi:10.48550/ARXIV.2202.09748
- [84] Xingya Zhao, Yixiong He, Liwen Huang, Junmin Mou, Ke Zhang, and Xiao Liu. 2022. Intelligent Collision Avoidance Method for Ships Based on COLRGes and Improved Velocity Obstacle Algorithm. *Applied Sciences* 12, 18 (2022), 8926. doi:10.3390/app12188926
- [85] Feixiang Zhu, Zhengyu Zhou, and Hongrui Lu. 2022. Randomly Testing an Autonomous Collision Avoidance System with Real-World Ship Encounter Scenario from AIS Data. *Journal of Marine Science and Engineering* 10, 11 (2022), 1588. doi:10.3390/jmse10111588

Received 2025-02-27; accepted 2025-03-31