

Automated Generation of Functionally Complete Assurance Suites for COLREGS-Compliance of Autonomous Surface Vehicles

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
daniel.varro@liu.se

Abstract

Autonomous surface vehicles (ASVs) operating in maritime traffic must comply with the International Regulations for Preventing Collisions at Sea (COLREGS). Demonstrating COLREGS compliance requires systematic assurance that all relevant encounter scenarios are represented and analyzed. Recent work has proposed a research agenda for scenario-based assurance of COLREGS compliance based on qualitative abstractions and multi-level scenario modeling. In this paper, we provide the first realization of a model-based approach that systematically derives initial scenes as assurance scenarios for multi-vessel encounters. First, a complete set of functional equivalence classes is derived automatically. Then, we provide a mapping from functional equivalence classes to logical scenarios that capture geometrical constraints between potentially hazardous ship encounters. Next, initial scenes with precise vessel placements are generated from logical scenarios by (1) a search-based algorithm and (2) a novel rejection sampling-based algorithm. Finally, we derive (static) trajectories amenable to simulation for ASV assurance purposes. Our evaluation demonstrates that the approach achieves full coverage of relevant COLREGS encounters at the functional level and scales to complex scenarios up to six vessels, which represent the vast majority of open sea encounters.

ACM Reference Format:

Dominik Frey, Ulf Kargén, and Dániel Varró. 2026. Automated Generation of Functionally Complete Assurance Suites for COLREGS-Compliance of Autonomous Surface Vehicles. In *ACM/IEEE 29th International Conference on Model Driven Engineering Languages and Systems (MODELS 2026)*, October 04–09, 2026, Málaga, Spain. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3822455.3830315>

1 Introduction

Independent and safe navigation is a central challenge for autonomous surface vehicles (ASVs). To operate in international waters, ASVs must comply with the International Regulations for Preventing Collisions at Sea (COLREGS) [16], a set of rules defined by the International Maritime Organization. These rules govern how the *own ship* (OS) should act to avoid collisions with others, the *target ships* (TSs). However, the COLREGS were designed for human interpretation, and they are intentionally underspecified, making their implementation in ASVs particularly difficult, especially in complex multi-vessel encounters.

In contrast to the autonomous road vehicles, where clear requirements on systematic safety assurance have already been formulated, e.g., in ISO 8800 [1], work on similar standards for the maritime domain is still ongoing, with the publication of a preliminary version of the so-called IMO MASS Code expected during 2026. However, the need for systematic assurance of COLREGS compliance has already been identified by maritime safety authorities [14, 69].

Inspired by safety assurance of autonomous road vehicles, five desiderata for assurance methods tailored to ASVs have been formulated in a recent paper [27]: (1) *Relevance*: an assurance suite should capture hazardous scenarios in terms of the COLREGS; (2) *Completeness*: an assurance suite should cover all semantically distinct COLREGS scenarios for a given number of vessels, including rare edge cases; (3) *Diversity*: the assurance suite should dedicate uniform attention to investigate each semantically distinct COLREGS scenarios; (4) *Scalability*: the approach must scale to larger numbers of vessels to represent complex encounters; (5) *Efficiency*: one must be able to generate the required assurance suite in reasonable time.

This recent study [27] also identified major semantic differences between the maritime and automotive domains (e.g., no road network, traffic rules solely defined based on relative bearings, evasive maneuvers in favor of braking, heterogeneous vessel types), and highlighted the limited cross-domain adaptability of assurance techniques developed originally for self-driving cars.

Problem statement. The goal of traditional testing approaches is to find bugs in ASV control software rather than providing *systematic assurance* of compliance. Several works rely on custom, often manually defined COLREGS-specific test scenarios evaluated in simulation or in field trials [3, 47, 60]. However, these methods do not ensure completeness [58]. Search-based and sampling-based scene generation methods [12, 13] improve automation and scalability, but by relying on stochastic exploration, they cannot guarantee completeness. Data-driven approaches based on historical maritime traffic logs [38, 62, 68] are restricted to observed traffic patterns, excluding plausible but unobserved scenarios, thus lacking completeness. Altogether, existing techniques do not provide assurance suites that are simultaneously relevant, diverse, complete, scalable, and efficient to generate. In particular, no existing approach addresses the critical requirement on *completeness*.

The research agenda in the vision paper [36] advocates multi-level test scenario synthesis where *functional equivalence classes* capture COLREGS encounters as graph models. Each equivalence class would then be refined into a set of geometric constraints. Next, an *initial scene* with concrete vessel placements could be obtained as a solution to this constraint problem. Finally, *trajectories* of target ships should be derived from the initial scenes. The vision paper



This work is licensed under a Creative Commons Attribution 4.0 International License. *MODELS 2026, Málaga, Spain*
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2809-9/2026/10
<https://doi.org/10.1145/3822455.3830315>

only provides solution for the generation of equivalence classes, while scene and trajectory generation is left for future work.

Objectives. This paper proposes a scalable scene generation technique based on *multi-step refinement (MSR)* to derive relevant, and (functionally) complete assurance suites for multi-vessel encounters. We operationalize the research agenda [36] to automatically derive initial scenes from a complete set of functional equivalence classes. Our goal is to systematically construct scenes which can be used by maritime safety authorities as an assurance suite to demonstrate COLREGS-compliance of ASV controllers. The specific contributions of the paper are as follows:

- (1) We introduce a novel approach to automatically derive assurance suites for COLREGS-compliance using refinement along multiple levels of abstraction.
- (2) We propose a new, more expressive functional scenario model based on geometrical relations between vessels.
- (3) We devise a constraint-based concretization approach using rejection sampling for deriving relevant initial scenes.
- (4) We provide a prototype implementation of the MSR approach.
- (5) We empirically validate the functional completeness, efficiency, and scalability of our prototype^{1 2}.

Added value. We provide a novel approach for automatically deriving assurance suites for validating COLREGS-compliance of autonomous vessels by synthesizing initial scenes and vessel trajectories. Our approach generates *relevant* scenarios by design by concretizing qualitative abstractions of COLREGS scenarios. The resulting assurance suites are *functionally complete* as we enumerate all structurally different equivalence classes exactly once and then concretize each into an initial scene. The approach is *scalable* and *efficient* for multi-ship encounters with up to six vessels (which represent the vast majority of cases in real maritime traffic [62]).

2 Background

2.1 Graph Models for Open Sea Encounters

Domain-specific graph models are frequently used to model complex cyber-physical systems [31, 43]. In this paper, graph models represent high-level *qualitative abstractions* of open-sea encounters.

Metamodel. A metamodel $M := \langle \Sigma, \alpha \rangle$ consists of a vocabulary $\Sigma := \mathcal{L} \cup \mathcal{R}$ and an arity function $\alpha : \Sigma \rightarrow \mathbb{N}$. Unary predicate symbols represent object roles within a scenario: $\mathcal{L} := \{OS, TS\}$, $\alpha(1) := 1$, for $1 \in \mathcal{L}$, where OS denotes the own ship (OS) (i.e., the system under test in our context), and TS denotes target ships (TS_i) representing maritime traffic. Binary predicate symbols represent relationships between objects: $\alpha(r) := 2$, for $r \in \mathcal{R}$.

Instance models. An instance model is a typed graph where each object (node) has a unique identity. Given a metamodel $M := \langle \Sigma, \alpha \rangle$, an instance model is a logic structure $F := \langle O_F, I_F \rangle \in \mathcal{F}_M$ 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: $I_F(r) : O_F^{\alpha(r)} \rightarrow \{\text{true}, \text{false}\}$. An instance model assigns a truth value to unary predicates over objects and binary relations over pairs of objects [8, 30].

¹Generated assurance suites available at <https://doi.org/10.5281/zenodo.20792733>

²Supporting research software available at <https://doi.org/10.5281/zenodo.19246756>

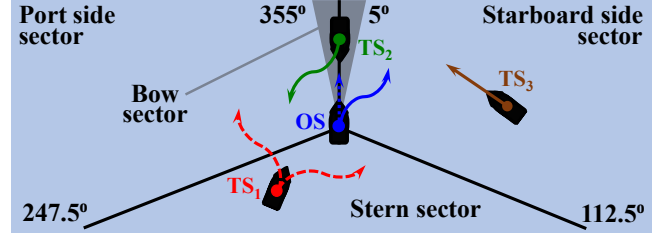


Figure 1: Running example: a four-vessel COLREGS scenario from [27] with illustrated bearing sectors.

We use a relational predicate syntax [52] to query the state of a model with $r \in \Sigma$, $\bar{o} \in O_F^{\alpha(r)} : r(\bar{o}) \equiv I_F(r)(\bar{o}) = \text{true}$, if r relation holds between objects in $\bar{o}$ and $\neg r(\bar{o}) \equiv I_F(r)(\bar{o}) = \text{false}$, if r relation does not hold between objects in $\bar{o}$. If $r(\bar{o})$ and $\neg r(\bar{o})$ hold at the same time, the model is *inconsistent*.

2.2 COLREGS Concepts and Situations

The COLREGS comprise 41 rules governing maritime navigation. We outline rules governing steering and sailing, which are the most relevant for ASV control software evaluation [63].

- **Rule 7: Collision risk** exists when two vessels maintain constant speed and heading on converging paths called *collision course*.
- **Rule 8: Action to avoid collision:** Vessels must keep "safe distance," often modeled as an elliptical *safety domain* [15, 48, 55], and make "readily apparent" maneuvers (course change $\geq 30^\circ$ [63]).
- **Rule 11: Deemed COLREGS situations:** Further rules are applicable when vessels are within *visibility range* [34].
- **Rule 22: Visibility range** is typically 1–6 nautical miles, depending on the vessels' length and relative bearing.

According to [16], a COLREGS *situation* is an open-sea encounter defined by the relative bearing between two vessels that are on a collision course within visibility range. Multiple such situations may coexist in a multi-vessel COLREGS *scenario*. Rules 13–15 classify these situations based on relative bearing sectors derived from the signal light angles defined in Rule 21. COLREGS define situation-specific maneuvers for each vessel involved in an encounter.

Example. Figure 1, adapted from [27], illustrates the sectors and corresponding maneuvers in a four-vessel scenario (serving as a running example of the paper) with the *own ship* (OS) and three *target ships* (TS_1 , TS_2 , and TS_3):

- (1) The *give-way* vessel must act early and decisively while the *stand-on* vessel maintains its course (Rules 16–17).
- (2) TS_1 *overtakes* OS to port or starboard if within its stern, and OS is within TS_1 's masthead sector. OS must give way (Rule 13).
- (3) TS_2 and OS are *head-on* if within each other's bow sectors. Both must give way to the starboard side (Rule 14).
- (4) OS is crossing TS_3 from port if OS is in TS_3 's port sector and TS_3 is within OS 's starboard sector. OS gives way (Rule 15).

Functional equivalence classes (FECs) of COLREGS. FECs are instances of graph models representing a COLREGS scenario. An initial metamodel $M_{\text{colr}} := \langle \mathcal{L}, \mathcal{R}_{\text{colr}} \rangle$ was introduced in [36], where

binary predicate symbols denote high-level COLREGS situation types between vessels: $\mathcal{R}_{\text{colr}} := \{\text{ovtkPort}, \text{ovtkStar}, \text{headOn}, \text{cross}\}$ representing situations (2) to port, (2) to starboard, (3), and (4) respectively. These relations allow to formally define a COLREGS situation predicate $\text{colreg}(o_i, o_j)$ as a *first-order logic* (FOL) expression $\text{ovtkPort}(o_i, o_j) \vee \text{ovtkStar}(o_i, o_j) \vee \text{headOn}(o_i, o_j) \vee \text{cross}(o_i, o_j)$ where $\langle o_i, o_j \rangle \in \mathcal{O}_F^2, o_i \neq o_j$.

Example. Figure 3 (left) shows a visual representation of an FEC for our COLREGS scenario where TS_1 overtakes OS to port (*ovtkPort*), while OS is in a *headOn* relation with TS_2 and crossing TS_3 (*cross*).

2.3 Assurance for COLREGS Compliance

COLREGS compliance of ASVs is often assessed by system-level testing [47, 58], where concrete encounter situations are executed in realistic simulators [3, 60]. In these approaches, COLREGS scenarios are typically treated as test cases aiming to reveal failures of ASV controllers to navigate challenging multi-vessel encounters.

A system-level test case usually comprises (1) an *initial scene* specifying precise positions, headings, and speeds for all vessels, which determines the subsequent interaction and its potential hazards [12]; (2) COLREGS-compliant *trajectories* for target ships, represented as time-series of positions and headings; and (3) a *test oracle* defining acceptable ASV behavior under the assumption that other vessels follow their prescribed paths. Executing such test cases in simulation provides test-based evidence of compliance for the ASV. The synthesis of initial scenes, trajectories, and test oracles are conceptually different challenges [9, 36].

Assurance suites. Safety standards typically necessitate structured arguments and supporting evidence constructed independently of a particular control software. We adopt the concept of *scenario-based assurance evidence* (from the automotive ISO 8800 standard [1]) for ASV encounters. Unlike a test suite, which primarily aims to detect bugs in ASV control software, an *assurance suite* needs to adequately represent *all* relevant and feasible COLREGS scenarios.

According to prior work [27, 58], a high-quality assurance suite should be *relevant*, targeting hazardous situations; *complete* and *diverse*, covering all semantically distinct encounter classes; while providing *scalable* (wrt. number of vessels) and *efficient* generation.

Relevant assurance suites. A relevant assurance scenario [27, 36] is defined as an encounter that satisfies the following criteria: (1) it involves a single ASV, (2) a certain number of target ships, and (3) each TS is engaged in a COLREGS situation with the OS . Additionally, (4) all TS s must adhere to the COLREGS. We enforce this by forbidding COLREGS situations between each TS pairs.

We use *COLREGS assurance constraints* to formally characterize relevant scenarios on the functional level. The constraints are formalized using FOL expressions over FECs. The formalizations are adopted from prior work [27] (and summarized in Table 1). The constraints are extended with further well-formedness rules to enforce *consistent models*, e.g., "*a vessel cannot be in a COLREGS relation with itself*" (for each vessel object o : $\neg \text{colreg}(o, o)$).

Functional completeness of assurance suites. A functionally complete assurance suite covers all semantically distinct COLREGS scenarios. Formally, semantically distinct scenarios are FECs with

unique graph shapes [50, 53]. A complete set of FECs consists of all structurally distinct graph shapes that can be instantiated from the metamodel M_{colr} and satisfy the COLREGS assurance constraints.

In this paper, our main focus is to provide a *novel method to automatically generate a functionally complete suite of initial scenes* for COLREGS-assurance. We also provide a proof-of-concept demonstrator for COLREGS-compliant *trajectory generation* to enable end-to-end simulation of ASV-agnostic test scenarios.

3 Multi-Step Refinement for ASV Assurance

As the first complete realization of the high-level agenda proposed in [36], we introduce a multi-step refinement (**MSR**) technique for generating a functionally complete set of assurance scenarios for validating COLREGS compliance. **MSR** systematically refines COLREGS assurance constraints through a sequence of automated refinement steps, as illustrated in Figure 2. The key novel contributions of our current work are highlighted in red in the figure (and discussed in depth in Section 4).

- §4.1 First, formal COLREGS assurance constraints are provided as input to a graph solver to automatically generate a complete set of unique graphs representing *functional equivalence classes* (FECs) [27]. FECs model relations between vessels on the COLREGS situation level, e.g., " TS_1 is overtaking OS ".
- §4.2 Next, we map each derived FEC to a *functional scenario model* (FSM), which is a novel, more expressive functional representation introduced in this paper precisely capturing the vessels' geometric relations, e.g., " TS_1 is in the stern sector of the OS ".
- §4.3 We introduce an automatic mapping from an FSM to a *logical scenario model* (LSM), representing each COLREGS situation as a numerical *constraint satisfaction problem* (CSP).
- §4.4 We provide techniques for automatically deriving *initial scenes* (ISs) from each LSM by solving the numerical constraints using an existing search-based method (**SB**) and a novel condition-driven sampling-based approach (**CDRS+PS**).
- §4.5 Finally, concrete COLREGS-compliant and simulation-ready trajectories are generated (statically) for target ships from ISs.

Conceptual novelty. Initial scene generation in [27] is based on manually defining an LSM and directly concretizing repeatedly without the guarantee of functional completeness. We refer to this approach as direct concretization (**DC**), which also serves as a baseline for our experiments later in Section 6.1. In contrast, **MSR** systematically enumerates all FECs and automatically constructs and solves LSMs to provide a *functionally complete assurance suite*.

Furthermore, we present a novel constraint-based method for initial scene generation using rejection sampling. Our method reduces the infeasible parts of the large sampling space imposed by the high degree of freedom in maritime encounters.

While maritime regulations and physical constraints make initial scene generation a unique, domain-specific challenge, we note that multi-step refinement has previously been proposed in the automotive domain for scene concretization [7]. Their approach samples functional models randomly; consequently, completeness at the functional level is not guaranteed. In contrast, our approach explores all FECs that satisfy the metamodel and COLREGS assurance constraints, thus ensuring full functional completeness.

Table 1: COLREGS assurance constraints formalized using FOL over functional models and COLREGS situations following [27].

Informal description	FOL expression for a $FEC \in \mathcal{F}_{FEC} \mid \forall \langle o, o_i, o_j \rangle \in \mathcal{O}_F^3, o_i \neq o_j$
(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 pair must be in a COLREGS situation.	$OS(o_i) \wedge TS(o_j) \implies (colreg(o_i, o_j) \vee colreg(o_j, o_i))$
(4) Two TS cannot be in a COLREGS situation.	$TS(o_i) \wedge TS(o_j) \implies \neg (colreg(o_i, o_j) \vee colreg(o_j, o_i))$

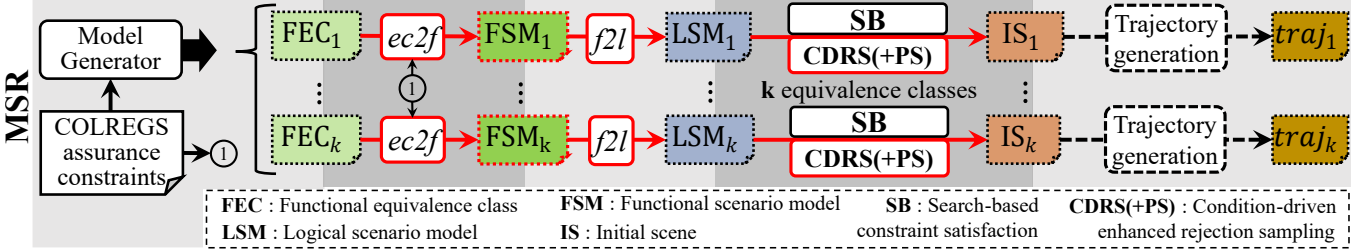


Figure 2: Overview of our multi-step refinement (MSR) with novel steps highlighted in red.

Table 2: Binary predicate symbols in M_{FSM} for FSMs.

Informal description	$r \in \mathcal{R}_{FSM}$
One vessel is in the other's stern / bow / port-side / starboard-side sector.	$inStern, inBow, inPort, inStar$
Two vessels are at / in / out of the visibility range.	$atVis, outVis, inVis$
Two vessels are on a collision course.	$mayCol$

4 Realization of MSR Approach

4.1 Generating Functional Equivalence Classes

We use graph generation techniques, following the approach outlined in [36] to derive a complete set of relevant FECs $\mathcal{T} \subset \mathcal{F}_{FEC}$, where each $FEC \in \mathcal{T}$ represents a semantically different assurance scenario (at the functional level). We use the Refinery solver [41] as graph generator. Refinery uses *partial models* [25] to generate a consistent and complete set of instance models.

As input for relevant FEC generation, we provide Refinery with M_{colr} as the metamodel and the COLREGS assurance constraints (from Table 1) as predicates using Refinery's syntax [42].

4.2 Deriving Functional Scenario Models

We introduce *functional scenario models* (FSMs), a new functional-level formalization of COLREGS scenarios that explicitly represents the geometric relations between vessels hidden in the COLREGS-specific relations captured by FECs. By refining FEC relations, FSMs enable the reuse of functional relations for trajectory generation without changing existing mappings. Furthermore, FSMs will support robustness testing by representing multiple geometric variations of the same FEC.

FSMs are instance models conforming to the metamodel M_{FSM} , which defines a set of binary relations, $\mathcal{R}_{FSM}$, listed in Table 2. These relations include relative bearing, visibility, and collision course.

To derive an FSM from an FEC, we define a rule-based mapping $ec2f: \mathcal{F}_{FEC} \rightarrow \mathcal{F}_{FSM}$, which translates each binary predicate over $\mathcal{R}_{colr}$ into a corresponding first-order logic formula over $\mathcal{R}_{FSM}$, as

detailed in Table 3. The mapping is derived directly from the semantics of FEC relations and is executed automatically. It preserves semantics by each generated FSM representing the same encounter situation as its corresponding FEC. Consequently, distinct FEC graph structures are translated into distinct FSM structures.

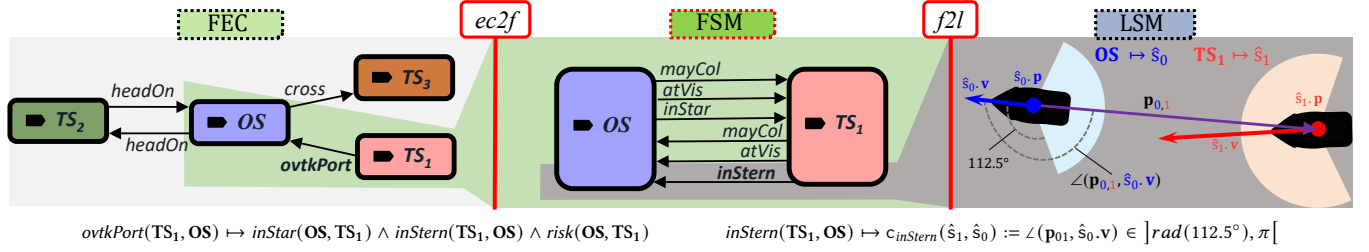
Example. From the FEC of our running example, $ec2f$ derives the corresponding FSM relations between objects shown in the middle column of Figure 3. In particular, the $ovtkPort$ relation between TS_1 and the OS implies that the two vessels are on a collision course ($mayCol$), and at visibility distance ($atVis$). Moreover, TS_1 is in the stern sector of the OS and the OS is in the starboard sector of TS_1 .

4.3 Mapping FSMs to Logical Scenario Models

In the next step, functional scenarios are transformed into CSPs. We introduce a mapping $f2l: \mathcal{F}_{FSM} \rightarrow \mathcal{N}$, where $\mathcal{N}$ is a set of *logical scenario models* (LSMs), each representing a CSP. The mapping $f2l$ automatically converts an FSM into its corresponding LSM. Functional-level objects are represented as numeric variables, and functional-level relations between objects are represented as binary geometric constraints over variables. The mapping uses parameters (distance thresholds, angles, etc.) acquired from the COLREGS and documented maritime best practices [63]. We define $f2l$ by

- (1) providing vessel attributes and area boundaries as external inputs (obtained from maritime standards and best practices);
- (2) mapping abstract objects $o_i \in \mathcal{O}_F$ in the FSM to a corresponding ship variable $\hat{s}_i$ with position, heading, length, and speed;
- (3) for every matching predicate $r_k \in \mathcal{R}_{FSM}$ in the FSM, we include a constraint c_k over the corresponding ship variables.

Example. Given the overtaking situation between $TS_1 \mapsto \hat{s}_1$ and the OS $\mapsto \hat{s}_0$, in the right column of Figure 3, the $f2l$ mapping is exemplified on the mapping $inStern \mapsto c_{inStern}$. They rely on relative bearing, such as $\angle(p_{0,1}, \hat{s}_0.v)$, which is the angular distance between the displacement $p_{0,1}$ and the velocity $\hat{s}_0.v$ vectors. $c_{inStern}$ ensures that the angle between $p_{0,1}$ and $\hat{s}_0.v$ is such that $\hat{s}_1.p$ is in the

Figure 3: $ec2f$ and $f2l$ mappings to derive numeric constraints from functional relations in the context of our running example.Table 3: Mapping between binary predicates in FECs and FSMs with $\langle o_i, o_j \rangle \in O_{FEC}^2 = O_{FSM}^2, o_i \neq o_j$.

Informal description	Predicates over $\mathcal{R}_{colr}$	Corresponding FOL expression over $\mathcal{R}_{FSM}$
Two vessels are at visibility distance and on a collision course.	$risk(o_i, o_j) \mapsto$	$atVis(o_i, o_j) \wedge atVis(o_j, o_i) \wedge mayCol(o_i, o_j) \wedge mayCol(o_j, o_i)$
The source vessel overtakes the target vessel on the port side.	$ovtkPort(o_i, o_j) \mapsto$	$inStar(o_j, o_i) \wedge inStern(o_i, o_j) \wedge risk(o_i, o_j)$
The source vessel overtakes the target vessel on the starboard side.	$ovtkStar(o_i, o_j) \mapsto$	$inPort(o_j, o_i) \wedge inStern(o_i, o_j) \wedge risk(o_i, o_j)$
A head-on situation between the source and target vessels.	$headOn(o_i, o_j) \mapsto$	$inBow(o_j, o_i) \wedge inBow(o_i, o_j) \wedge risk(o_i, o_j)$
The source vessel crosses the target vessel from the port side.	$cross(o_i, o_j) \mapsto$	$inStar(o_j, o_i) \wedge inPort(o_i, o_j) \wedge \neg headOn(o_i, o_j) \wedge risk(o_i, o_j)$

stern sector of the OS. We map the rest of the relative bearing relations similarly ($inBow \mapsto c_{inBow}$, $inPort \mapsto c_{inPort}$, $inStar \mapsto c_{inStar}$), along with *visibility distance constraints* and *near collision constraints* ($atVis \mapsto c_{atVis}$, $outVis \mapsto c_{outVis}$, $inVis \mapsto c_{inVis}$, $mayCol \mapsto c_{mayCol}$). The full set of formulae used in $f2l$ is explained in [27].

4.4 Deriving Initial Scenes

An *initial scene* (IS) is the first snapshot of a vessel encounter, defined by the precise position, speed, course, and size of each vessel. In the LSM, these attributes are modeled as free variables. An IS is a concrete assignment to these variables that satisfies all LSM constraints. A concrete IS to our running example scenario (reduced to three vessels) is shown in Figure 4. To derive ISs, we treat the LSMs as numerical CSPs and apply constraint satisfaction techniques to compute satisfying variable assignments. We propose two different approaches for solving the CSP problem. The first is the search-based method **SB** adapted from prior work [27], applied directly to our LSMs. The second is a *novel condition-driven sampling strategy* (CDRS), which provides an efficient sampling mechanism by resolving conditional dependencies between LSM constraints using a sequenced sampling procedure. The details of the two constraint satisfaction approaches are described in Section 5.

4.5 Trajectory Generation and Simulation

Generating TS trajectories from initial scenes is a challenging problem [44, 56] in itself. In a real-world COLREGS encounter, ship captains can adjust their courses based on observed actions by other vessels. Since our approach generates trajectories statically from ISs (i.e., the path of each TS is predetermined at the start of simulation), we need to minimize the risk that trajectories become non-compliant because of unexpected actions by the ASV.

We have implemented a proof-of-concept demonstrator of COLREGS-compliant trajectory generation based on the Rapidly-exploring Random Tree Star (RRT*) algorithm [23]. Trajectories are generated under the assumptions that (1) each TS assumes that the OS will act fully COLREGS-compliant relative to the TS and

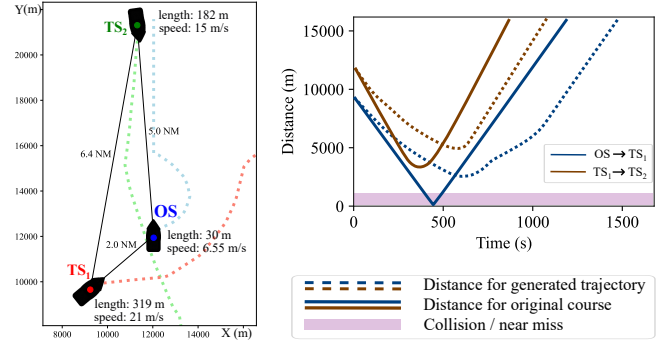


Figure 4: Simulation result of a hazardous scenario.

(2) all TSs will act COLREGS compliant relative to each other. These assumptions align with existing works [37, 58].

As an early validation of simulatability, we integrate the generated trajectories into the simulation environment of the WARA-PS Core System [4], which is a complex, multi-domain industry-driven research arena designed to support integration, demonstration, and testing of autonomous systems for public safety.

Example. Figure 4 (left side) shows trajectories generated for our sample COLREGS scenario after simulation. The right side shows inter-vessel distance over time, comparing the generated trajectories with the case where all vessels maintain their original courses. Our approach ensures that, initially, TS1 and TS2 are not on a collision course with each other while OS is on a collision course with both TS1 and TS2. Hence, the situation is potentially dangerous without evasive action. Our simulation demonstrates that the vessels avoid collision and near misses along the generated trajectories.

Current limitations. Trajectory generation does not currently take the “readily apparent” constraint (COLREGS rule 8) into account. Future work will address this limitation and improve trajectory realism by incorporating vessel dynamics. We also plan to investigate

an alternative approach based on on-the-fly (online) simulation feedback to iteratively refine trajectories.

5 Solving Logical Scenario Constraints

This section provides a detailed discussion on how we derive an initial scene, i.e., a solution to the CSP representing the logical scenario model, by using (a) a search-based method from [27], and (b) a novel method using *condition-driven* rejection sampling.

Definition [27]. An LSM is a tuple $N := \langle S_N, C_N, m_N, B_N \rangle$ where: S_N is a finite set of ships, each represented as an oriented point (see below); C_N is a set of binary numerical constraints $c_k(\hat{s}_i, \hat{s}_j)$ over ships $\hat{s}_i, \hat{s}_j \in S_N$; m_N is a sea map bounding the positions of ships in S_N ; B_N is a function specifying bounds on ship attributes: $B_N(\hat{s}_i) := \langle \text{LMin}_i, \text{LMax}_i, \text{spMin}_i, \text{spMax}_i \rangle \in \mathbb{R}^4$.

Each ship $\hat{s}_i \in S_N$ is represented by a tuple $\hat{s}_i := \langle x, y, h, l, \text{sp} \rangle$, where $x, y \in \mathbb{R}$ denote the ship's center position; $h \in \mathbb{R}$ is the heading angle (in radians); $l \in \mathbb{R}$ is the ship's length (in meters); $\text{sp} \in \mathbb{R}$ is the speed (in m/s). In addition, we derive position $\mathbf{p} := (x, y)$ and velocity $\mathbf{v} := (\cos h, \sin h) \times \text{sp}$ vector attributes.

Based on [27], an IS is a solution scene $_N : S_N \rightarrow \mathbb{R}^5$ to the LSM N , assigning concrete values to each ship's variables such that:

- (1) Each ship's position and attributes lie within bounds:
 - scene $_N(\hat{s}_i).x \in [0, m_N.x]$, scene $_N(\hat{s}_i).y \in [0, m_N.y]$;
 - scene $_N(\hat{s}_i).h \in [-\pi, \pi]$;
 - scene $_N(\hat{s}_i).l \in [B_N(\hat{s}_i).\text{LMin}, B_N(\hat{s}_i).\text{LMax}]$;
 - scene $_N(\hat{s}_i).\text{sp} \in [B_N(\hat{s}_i).\text{spMin}, B_N(\hat{s}_i).\text{spMax}]$;
- (2) All pairwise constraints $c_k(\hat{s}_i, \hat{s}_j) \in C_N$ are satisfied.

5.1 Search-Based Constraint Satisfaction

To provide a solution to N , we apply convergence-focused many-objective search **SB**, handling N as a general *optimization problem* (OP) [32]. The method is adapted from modern search-based scene generation in maritime [27] and autonomous driving [2, 29, 46] contexts based on *genetic search* [19].

To define the many-objective OP derived from the numeric problem N , let Π be a penalty function assigning a non-negative violation degree to each constraint in a candidate scene. For each constraint $c \in C_N$, let actors(c) $\subseteq S_N$ denote the ships involved in c . For each ship $\hat{s}_i \in S_N$, an objective function f_i is defined by $f_i(\text{scene}) := \sum_{c \in C_N, \hat{s}_i \in \text{actors}(c)} \Pi(c, \text{scene})$. The resulting many-objective OP is $\min(f_1(\text{scene}), \dots, f_{|S_N|}(\text{scene}))$. A candidate scene is feasible for N iff $\forall i \in \{1, \dots, M\} : f_i(\text{scene}) = 0$. Solver execution terminates once such a scene is found.

Since the OPs in **MSR** are many-objective, we use NSGA3 [18, 35] as the solver. Its hyperparameters are systematically tuned through a grid search procedure, following established best practices in evolutionary algorithm configuration [24]. The final configuration uses a *population size of 30*, *simulated binary crossover* [20] with probability 1.0 and distribution index $\eta = 5$, and *polynomial mutation* [20] with probability 1.0 and $\eta = 15$. Selection is performed using binary tournament with fitness domination. *Reference points* are generated using the *Das & Dennis method* [17] with one partition, which is advised for single-point Pareto fronts [22] (note that a feasible solution has a penalty of 0).

5.2 Condition-Driven Rejection Sampling

In its basic form, rejection sampling (**RS**) [28] introduced for maritime in [27] works by drawing every TS's position, heading, length, and speed uniformly from their *domains regions* (m_N and B_N). The position region is constrained by the possible visibility distances (2, 3, 5, or 6 NM) defined by the COLREGS (see Rule 22 in Section 2.2). The velocity region is constrained by a *velocity obstacle cone* (VOC) [21], enforcing a collision course for the **OS** and **TS**s. After sampling, all constraints (see the examples in Section 4.3) are evaluated on the scene: if every constraint is satisfied, the sample is accepted; otherwise, it is discarded, and the process repeats. Although this approach guarantees correctness, it is inefficient when the solution space is only a small fragment of the sampling regions.

5.2.1 Rejection sampling workflow. Next, we describe **RS** in detail.

Step 1: Defining sampling regions. In this step, we construct the two-dimensional regions from which we sample the ship attributes. The **OS**'s attributes are fixed, user-configurable parameters (position $\mathbf{p}_0$, velocity $\mathbf{v}_0$, length $\hat{s}_0.l$). We distinguish two types of sampling regions for each **TS** $_i$: a *position region* ($\mathcal{A}_{\text{pos},i}$) to sample $\mathbf{p}_i$ and a *velocity region* ($\mathcal{A}_{\text{vel},i}$) to sample $\mathbf{v}_i$.

For position regions, which need to satisfy constraints like c_{atVis} , c_{inVis} , and c_{outVis} , we calculate all the possible pairwise visibility distances between the **OS** and **TS** $_i$ based on COLREGS Rule 22 and define $\mathcal{A}_{\text{pos},i} := \mathcal{M}_i \cap \mathcal{D}_i$, where:

- $\mathcal{M}_i = \{\mathbf{p} \mid (0, 0) \leq \mathbf{p} \leq (m_N.x, m_N.y)\}$ is region of the map;
- $\mathcal{D}_i$ is a union of circular regions centered at $\mathbf{p}_0$ with radii equal to all the possible visibility distances between **OS** and **TS** $_i$;

The velocity region is defined as: $\mathcal{A}_{\text{vel},i} := \mathcal{S}_i \cap \mathcal{V}_i$, where:

- $\mathcal{S}_i := \{\mathbf{v} \mid \|\mathbf{v}\| \in [B_N(\hat{s}_i).\text{spMin}, B_N(\hat{s}_i).\text{spMax}]\}$ is the speed range constrained by B_N ;
- $\mathcal{V}_i$ is a VOC region enforcing collision course with the **OS**.

Step 2: Constrained sampling and assignment. Next, **TS** $_i$ position and velocity vectors are sampled uniformly from sampling regions $\mathcal{A}_{\text{pos},i}$ and $\mathcal{A}_{\text{vel},i}$. Furthermore, attribute is sampled from the corresponding bounds in $B_N(\hat{s}_i)$. Formally: $\hat{s}_i.\mathbf{p} \sim \mathcal{U}(\mathcal{A}_{\text{pos},i})$, $\hat{s}_i.\mathbf{v} \sim \mathcal{U}(\mathcal{A}_{\text{vel},i})$, $\hat{s}_i.l \sim \mathcal{U}([B_N(\hat{s}_i).\text{LMin}, B_N(\hat{s}_i).\text{LMax}])$ where $\mathcal{U}$ is the *normalized Lebesgue measure* over regions. Finally, we assign the sampled values to scene($\hat{s}_i$).

Step 3: Rejection phase. Numerical constraints between **TS**s may introduce complex circular and cross-dependencies that are impossible to express with sampling regions [7, 27]. Instead, we over-approximate sampling regions and use rejection sampling by checking whether the sampled scene satisfies all pairwise constraints $c_k(\hat{s}_i, \hat{s}_j) \in C_N$. If so, the scene is accepted as a valid scene $_N$ to N ; otherwise, the sample is discarded, and we jump to Step 1.

5.2.2 Novel strategies. Our *new rejection sampling* strategy improves on **RS** with two enhancements: (1) *length pre-sampling (PS)*, which samples ship lengths in advance to tighten visibility distance bounds, and (2) *condition-driven sampling (CDRS)*, which uses LSM-defined relative bearings to constrain sampling regions. We denote the combination of these two methods as **CDRS+PS**.

Length pre-sampling enhancement (PS). As an enhancement to the original **RS** workflow, we constrain each visibility region $\mathcal{D}_i$ by

sampling the ships' length independently before Step 1. As a result, we can derive a smaller set of pairwise visibility distances based on COLREGS Rule 22 as a function of scene($\hat{s}_0$).1 and scene($\hat{s}_i$).1, resulting in fewer invalid scenes.

Condition-driven sampling enhancement (CDRS). To further enhance sampling, we exploit additional information about the relative bearing of vessels contained in the LSMs of **MSR**. This is achieved (1) by constraining pairwise visibility distance regions $\mathcal{D}_i$ based on COLREGS Rule 22 as a function of visibility angle, and (2) by further limiting the position $\mathcal{A}_{pos,i}$ and velocity $\mathcal{A}_{vel,i}$ regions to the intersection with conical sectors corresponding to allowable bearing ranges in the LSM ($c_{inStern}$, c_{inBow} , c_{inPort} , c_{inStar}).

For instance, if $c_{inStern}(\hat{s}_i, \hat{s}_0)$ holds, then it defines a 135° -wide cone aligned with $-\mathbf{v}_0$, shown in the right column of Figure 3, representing the stern sector of the **OS**. $\mathbf{TS}_i$ must be sampled from within this cone region. Other bearing-based constraints are defined analogously using their respective angular cones.

Combined approach (CDRS+PS). Based on the pre-sampled lengths and relative bearing sectors derived from the FSM, we can calculate the exact visibility distances. This condition-driven enhanced rejection sampling ensures that candidate scenes always satisfy visibility constraints between the **OS** and **TSs** informed by ship sizes and FSM-defined relationships, reducing rejection rates substantially compared to the standard **RS**.

As the underlying sampling framework, we chose *Scenic* [26, 61], a probabilistic programming language for modeling environments using high-level constraints. *Scenic* also provides the necessary geometric sampling regions (e.g., circular, conical) and operations over regions (e.g., $\cup$, $\cap$). It is used for problems both in the automotive [7] and the maritime domain [27]. We implement **PS** as an individual sampling step before a problem is defined for *Scenic*. Then, we dynamically generate *Scenic* code based on the input, **CDRS**, and Step 1. We run one sampling iteration with *Scenic* (Step 2) and check constraint satisfaction (Step 3). This procedure is repeated until a valid solution is found.

6 Evaluation

Our evaluation aims to determine to what extent **MSR** satisfies the five desiderata for an effective assurance suite defined in [27] (see Section 1). By construction, **MSR** guarantees 100% *relevance* (thanks to the COLREGS assurance constraints), and *functional-level diversity* (as we generate exactly one test case per FEC). As such, our experimental evaluation will focus on assessing *functional completeness*, *scalability* and *efficiency* of assurance scene generation. Specifically, we investigate the following research questions:

- **RQ1:** How complete are assurance suites generated by **MSR**?
- **RQ2:** How scalable and efficient is **MSR**?

Additionally, we conduct an ablation study (**AB**) to evaluate the individual impacts of the proposed improvements to the **RS** constraint satisfaction method.

6.1 A Baseline for Assurance Scene Generation

Assurance scene generation approaches in maritime can be broadly categorized into (1) search-based generation, (2) sampling-based

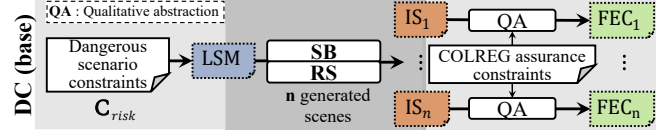


Figure 5: Baseline approach: direct concretization (DC).

generation, and (3) extraction of scenarios from historical encounters. The latter has previously been shown to be insufficient for deriving relevant multi-vessel assurance scenes [27] and is therefore not considered further. We compare against direct concretization (**DC**), which implements abstraction-based synthetic scene generation using metaheuristic search (**SB**) and rejection sampling (**RS**) [27]. Since **SB** uses NSGA2 [19], which is suboptimal for the many-objective optimization problem, we additionally evaluate an NSGA3 variant. While openly available implementations are limited, **DC** is publicly available and fully reproducible.

As shown in Figure 5, **DC** formalizes *dangerous scenario constraints* as a *single* joint LSM over vessels. The LSM is formulated as a CSP, where vessels are represented with numerical variables, and their geometric relations as numerical constraints: $C_{risk} := \{c_k(\hat{s}_i, \hat{s}_j) \mid i, j \in \{0 \dots n\}\}$, where

$$c_k := \begin{cases} c_{atVis}(\hat{s}_i, \hat{s}_j) \wedge c_{mayCol}(\hat{s}_i, \hat{s}_j) & \text{if } i = 0, j > 0 \\ c_{outVis}(\hat{s}_i, \hat{s}_j) \vee \neg c_{mayCol}(\hat{s}_i, \hat{s}_j) & \text{if } i, j > 0, i \neq j \end{cases}$$

Informally, the own ship ($\hat{s}_0$) is on a collision course and at visibility range with each target ship ($\hat{s}_i \mid i > 0$). No two target ships within visibility range are on a collision course.

DC derives a set of initial scenes by directly solving C_{risk} (without FEC guidance as in **MSR**). To assess the functional completeness of the assurance suite, the initial scenes are abstracted into FECs using qualitative abstraction over graph shapes. The abstraction uses COLREGS assurance constraints (see Section 2.3) to identify functional relations between vessels and derive relevant FECs. Previous results indicate that **DC** runs efficiently, but provides no functional completeness guarantees for the generated set of initial scenes [27].

6.2 Measurement Setup

Compared approaches. We automatically derive initial scenes using the **MSR** approach and compare the results with those generated using the **DC** approach. We run the two approaches in various configurations (see Table 4) using the constraint satisfaction methods introduced in Section 5. The first three rows contain the baselines, e.g., **DC/SB3** denotes the **DC** approach using NSGA3 as logical constraint solver. Note that **CDRS** and **CDRS+PS** are only compatible with the **MSR** approach, as they depend on the existence of FSMs.

Metrics. To measure functional completeness for **RQ1**, we use the same coverage metric as in [27] and calculate the percentage of covered unique relevant FECs among all possible relevant FECs in an assurance suite: $\frac{|\text{covered FECs in } \mathcal{T}|}{|\text{shapes in } \mathcal{T}|} \times 100$. For **RQ2**, we measure scalability and efficiency by calculating cumulative coverage at given timestamps throughout the generation of an assurance suite. We also include a min-max interval at each time step. Moreover, we calculate the average time to cover a unique relevant FEC: $\frac{\text{total generation time}}{|\text{covered unique FECs in } \mathcal{T} \text{ at the end of the generation}|}$.

Table 4: Configurations of compared approaches.

Measurement ID	Approach	Constr. sat. / engine	Compared in
DC/RS as in [27]	DC	RS	RQ1, RQ2, AB
DC/SB2 as in [27]	DC	SB with NSGA2	RQ1, RQ2
DC/SB3	DC	SB with NSGA3	RQ1, RQ2
DC/RS+PS	DC	RS+PS	AB
MSR/RS	MSR	CDRS	AB
MSR/RS+PS	MSR	CDRS+PS	RQ1, RQ2, AB
MSR/SB3	MSR	SB with NSGA3	RQ1, RQ2

Table 5: Input parameters provided for the LSM $\in \mathcal{N}$. The units are in meters (m), knots (kn), and nautical miles (NM).

$i = 0 \dots n-1 :$	$i = 0$ for OS	$i > 0$ for TSs
$[B_N(\hat{s}_i).lMin, B_N(\hat{s}_i).lMax]$	[30 m, 30 m]	[10 m, 400 m]
$[B_N(\hat{s}_i).spMin, B_N(\hat{s}_i).spMax]$	[0 kn, 50 kn]	
$[m_N.x, m_N.y]$	[13 NM, 13 NM]	

Table 6: Total number of FECs and global timeouts.

n	2	3	4	5	6
Number of FECs	7	28	84	210	462
Global timeout	28 min	112 min	336 min	840 min	1848 min

Scene generation setup. In this paper, we investigate COLREGS scenarios involving 2–6 vessels, as relevant encounters involving more than six vessels are extremely rare in open-sea settings [62]. We refer to the number of vessels in a scenario as $n = |\mathcal{S}_N|$. The input parameters for the LSMs $\in \mathcal{N}$ are detailed in Table 5, reflecting characteristics of small to large-sized ships.

In a maritime assurance setting, a key quality metric of a scenario generator is the time required to generate a (functionally) complete set of semantically distinct scenarios. Therefore, we ensure that all compared approaches have the same time budget for a given number of vessels. Moreover, since metaheuristic search might get stuck in local optima [39], we enforce a *per-scene timeout* t_S of 4 minutes. This timeout is based on empirical observations in the 6-vessel case using a genetic search, balancing sufficient time for convergence against early termination when stuck in local optima. The *global timeout* for each vessel count is calculated by multiplying the total number of FECs by t_S . This setup ensures that each measurement allocates an average of 4 minutes to each equivalence class across all configurations. The total number of FECs and corresponding global timeouts for each case (2–6 vessels) are shown in Table 6.

For the **MSR** approaches, we attempt to cover all FECs, allocating up to t_S minutes per attempt. If all FECs have been attempted before the global timeout, we revisit uncovered ones until either all FECs are covered or the global timeout is reached.

For **DC/SB2** and **DC/SB3**, the search is not targeted towards a specific FEC. Therefore, we repeatedly run the search for t_S until the global timeout. For **DC/RS** and **DC/RS+PS**, only the global timeout is used, as local optima are not an issue for sampling-based approaches. Finally, we abstract the concrete scenes for each *batch* (i.e., particular approach and number of vessels) into FSMs by building the corresponding LSM (see [27]) and applying the reverse

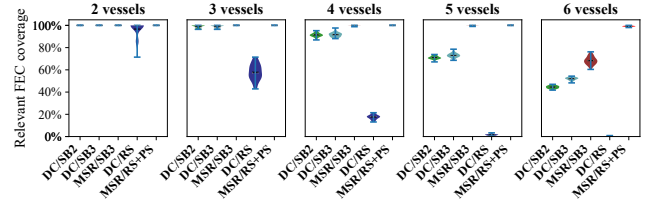


Figure 6: Unique relevant FEC coverage distribution.

Table 7: Median of unique relevant FEC coverage (top) and median of average times per unique relevant FEC (bottom).

	Approach	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$
Coverage	DC/SB2	100%	100%	91.67%	70.95%	44.48%
	DC/SB3	100%	100%	91.67%	72.85%	52.16%
	MSR/SB3	100%	100%	100%	100%	68.18%
	DC/RS	100%	57.14%	7.85%	1.42%	0%
	MSR/RS+PS	100%	100%	100%	100%	99.13%
Average time / FEC	DC/SB2	2.7 s	142.5 s	261.8 s	338.3 s	539.6 s
	DC/SB3	2.4 s	198.7 s	261.8 s	329.4 s	460 s
	MSR/SB3	0.43 s	21.6 s	49.8 s	187.9 s	351.6 s
	DC/RS	92.4 s	420 s	1344 s	16800 s	timeout
	MSR/RS+PS	0.17 s	0.76 s	4.9 s	30.2 s	241.8 s

mapping of $f2l$. The resulting FSMs are further abstracted into FECs by using the inverse of $ec2f$.

Measurement configuration. Due to the stochastic nature of search-based and sampling-based algorithms, we ran our measurements 30 times for every approach and vessel number with different random seeds. The random seeds are fixed to guarantee reproducibility. For stability, the garbage collector is explicitly invoked, and two warm-up runs are executed before each measurement batch. The experiments are carried out on Ubuntu 24.04 with a 6th Gen Intel Haswell CPU (model 60, family 6), 32 GB RAM, and Python 3.12.

We perform statistical analysis to compare the resulting data sets for each batch using the non-parametric *Mann–Whitney U test* [40] to compute p-values, and *Vargha–Delaney* (A_{12}) [5] for effect sizes.

6.3 RQ1: How complete are assurance suites generated by MSR?

Motivation. Two scenarios may differ on a concrete scene level but have the same semantics on the FEC level. Ideally, all unique relevant FECs should be covered with at least one concrete initial scene. **RQ1** investigates to what extent the compared approaches cover semantically distinct, relevant scenes.

Results. Figure 6 shows the distribution of relevant FEC coverages (across all 30 runs) for each batch of **DC/SB2**, **DC/SB3**, **MSR/SB3**, **DC/RS**, and **MSR/RS+PS** within the global timeout. The median values are listed at the top of Table 7.

DC/SB2 has a coverage decreasing from 100% ($n = 2$) to ~45% ($n = 6$), while **MSR/SB3** drops down only to ~68% for $n = 6$.

DC/SB2 and **MSR/SB3** have a 100% coverage for $n = 2$; however, for $n \geq 3$, the differences are statistically significant (p-values $\ll 0.001$) and increasingly large (with effect sizes of 0.7 for $n = 3$, 1.0

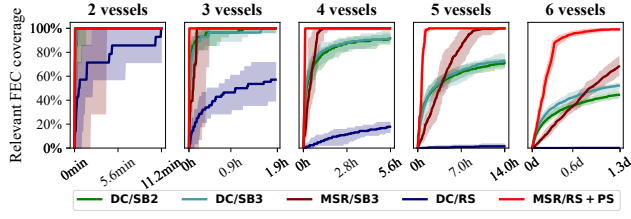


Figure 7: Cumulative coverage of unique relevant FECs.

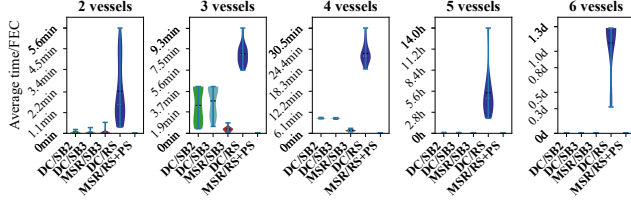


Figure 8: Average times per unique relevant FEC.

for $n > 3$). This suggests that **MSR/SB3** increasingly outperforms **DC/SB2**. A similar trend is observed for **DC/SB3**.

For sampling-based approaches, **DC/RS** has the lowest coverage decreasing from $\sim 100\%$ ($n = 2$) to 0% ($n = 6$) while **MSR/RS+PS** yields 100% coverage for every seed up to $n = 5$. For $n = 6$, some seeds of **MSR/RS+PS** reach 100% coverage in 1.1 days, while the median coverage was 99% within the global timeout.

DC/RS vs. **MSR/RS+PS** exhibits a strong and significant (p-values $\ll 0.001$) difference across all vessel counts. For $n = 2$, the effect size is moderate (0.6), but rapidly escalates: 0.7 for $n = 3$, and 1.0 for $n > 3$, suggesting that **MSR/RS+PS** outperforms **DC/RS** by far.

RQ1: **MSR/SB3** and **MSR/RS+PS** consistently achieve higher FEC coverage than their respective baselines as the number of vessels increases. They maintain steady 100% coverage across all vessel counts up to $n = 5$ and **MSR/RS+PS** is the only approach with runs reaching 100% coverage for $n = 6$ within the timeout.

6.4 RQ2: How scalable and efficient is MSR?

Motivation. Assuring ASV COLREGS compliance requires a large number of initial scenes, making generation efficiency and scalability essential. **RQ2** compares **MSR** and the baseline in coverage growth and scalability as vessel numbers increase.

Results. Figure 7 shows the cumulative coverage over time (with min-max intervals), and Figure 8 shows the distribution of average times (across all 30 runs) to cover a unique relevant FEC for each batch of **DC/SB2**, **DC/SB3**, **MSR/SB3**, **DC/RS**, and **MSR/RS+PS**. The median values are listed at the bottom of Table 7.

For search-based approaches, **MSR/SB3** consistently covers a unique FEC faster than **DC/SB3** and **DC/SB2**. The median coverage times (in sec) are 0.43 vs. 2.4–2.7 for $n = 2$, and 351.6 vs. 460–539 for $n = 6$. The statistical analysis of **DC/SB3** vs. **MSR/SB3** supports this with effect sizes ranging from 0.85 to 1.0 as the number of vessels increases, also indicating a growing performance gap under more complex scenarios with statistical significance (p-values $\ll 0.001$).

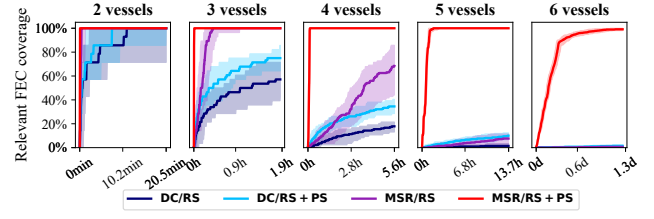


Figure 9: Cumulative coverage of unique relevant FECs for sampling-based approaches.

For sampling-based approaches, **MSR/RS+PS** consistently outperforms **DC/RS** in terms of coverage speed with effect sizes of 1.0 and p-values of $\ll 0.001$: median times (in sec) are 0.17 vs. 92.4 for $n = 2$, and 241.8 vs. *timeout* for $n = 6$, since **DC/RS** fails to cover any FEC within the global timeout (1848 minutes).

Since only **MSR/RS+PS** and **MSR/SB3** scale to $n = 6$ based on the cumulative coverage, we additionally measure the time they take to reach 100% coverage beyond the global timeout. For **MSR/RS+PS**, the median time (across all 30 seeds) to full functional coverage is ~ 2 days, and for **MSR/SB2**, it is ~ 3.8 days.

RQ2: Both **MSR** approaches consistently achieve faster coverage growth and better scalability with increasing number of vessels compared to baselines. Only **MSR** approaches scale to $n = 6$ with **MSR/RS+PS** reaching steady 100% coverage within 2 days.

6.5 AB: Ablation Study

Measurement setup. We evaluate the individual and combined effects of the proposed enhancements (**PS** and **CDRS**) to sampling-based constraint satisfaction. We compare (1) the original **DC/RS**, (2) **DC/RS** enhanced with pre-sampling (**DC/RS+PS**), (3) **MSR** with condition-driven sampling but *without* pre-sampling (**MSR/RS**), and (4) **MSR** using *both* condition-driven sampling and pre-sampling (**MSR/RS+PS**). (Recall that condition-driven sampling is not applicable for direct concretization since it requires FSMs.)

To measure the improvement in scalability and efficiency, we use the cumulative coverage of unique relevant FECs, which indicates both total coverage and speed.

Results. Figure 9 presents the cumulative coverage over time for each batch of **DC/RS**, **DC/RS+PS**, **MSR/RS**, and **MSR/RS+PS**. The results indicate that both **DC/RS+PS** and **MSR/RS** can individually improve performance compared to **DC/RS** up to $n = 4$. Notably, for $n = 3$, **MSR/RS** achieves 100% coverage within ~ 30 minutes. For $n = 4$, the base coverage of **DC/RS** ($\sim 18\%$) increases to $\sim 35\%$ with **DC/RS+PS** and to $\sim 68\%$ with **MSR/RS**.

However, for $n > 4$, neither enhancement alone is sufficient to achieve significant improvement, resulting in total coverage below 10%. In contrast, the combined approach, **MSR/RS+PS**, attains steady 100% coverage up to $n = 5$ and reaches 100% for $n = 6$.

In terms of efficiency, we observe statistically significant differences in the average time per covered FEC between **DC/RS** and both **DC/RS+PS** and **MSR/RS**. Specifically, for all n except $n = 2$, the improvements yield p-values < 0.001 and large effect sizes (1.0). For $n = 2$, **DC/RS+PS** is only slightly faster than **DC/RS**.

AB: Both pre-sampling (**DC/RS+PS**) and condition-driven sampling (**MSR/RS**) individually yield significant improvements over the baseline (**DC/RS**) for $n \leq 4$. However, in order to scale beyond $n = 4$, combining the two (**MSR/RS+PS**) is necessary.

6.6 Threats to Validity

Internal validity. We handle the stochastic nature of search-based and sampling-based algorithms in compliance with best practices of statistical evaluations. The 4-minute timeout may pose a threat to internal validity, as we can get different results with a longer or shorter timeout. This duration was chosen based on empirical observations from complex 6-vessel scenarios using genetic search, where the average convergence time is well below 4 minutes.

External validity. Our investigations are limited to relevant scenarios of 2–6 vessels identified in related works [27, 36] for validating COLREGS compliance. However, our method is extensible, thus allowing to incorporate additional constraints over FECs as input.

One may observe different results between search-based and sampling-based concretization methods when scaling to scenarios with more than six vessels, or when incorporating more complex environmental geometry (e.g., shorelines) or other vessel types (e.g., sailing boats, stationary vessels).

6.7 Discussion

MSR can achieve full coverage including rare scenarios. **MSR** finds a corresponding initial scene for every FEC. In the 6-vessel setting, out of the 462 equivalence classes, approximately 15% were covered only by **MSR** among the compared approaches within the given timeout (even after repeating the experiments with 30 different seeds). One of such cases is where the **OS** is simultaneously overtaken by five **TSs**. This results in a hard-to-solve constraint set due to the proximity of the **TSs** and the necessitated absence of COLREGS situations between them. Covering such rare critical cases is important from a safety assurance perspective, but such sparse solution spaces are problematic for direct concretization.

MSR is efficient and scalable up to six vessels. The average computational time of generating a unique relevant FEC using **MSR** is consistently lower than that of the baselines. For each run, **MSR** systematically restricts the search space to a constrained region characterized by a single FEC. On the other hand, **DC** may fail to find a solution (if that fragment of the solution space is sparse), or it may find too many solutions in a fragment (if the particular fragment is dense with solutions), which negatively affects efficiency.

Scene-level diversity. To gain extra insights, we further analyzed the initial scenes generated by **MSR/RS+PS** and **MSR/SB3**. COLREGS situations from **MSR/RS+PS** are easy to identify, whereas **MSR/SB3** provides boundary cases where ship captains or ASV controllers are more likely to misidentify a particular situation. Intuitively, small changes in vessel position or heading can alter the COLREGS classification. Since **MSR/SB3** uses genetic search as a constraint satisfaction engine, it stops when the first feasible solution is reached within the boundary (numerical limits), hence the retrieved solutions remain near those boundaries (as mutations are more likely to be local). By applying **MSR/SB3** for boundary

value analysis and **MSR/RS+PS** for nominal value testing, the two methods can be combined to address *scene-level* diversity within a particular FEC to facilitate robustness testing of ASV controllers (which has not been discussed in previous work [27]). However, this is beyond the scope of the current work.

Transferability. For richer maritime scenarios (e.g., static obstacles, vessel classes, physical environments), FECs can be extended with new relation symbols and constraints. New FEC relations may either map to new FSM relations or reuse existing ones. Each new FSM relation requires a corresponding logical constraint with realistic numeric boundaries derived from vessel characteristics. Since FECs and FSMs are finitely enumerable abstractions, the conceptual generalizability of our approach depends only on the satisfiability of the extended FSMs. However, our empirical evidence of functional completeness is limited to the settings evaluated in this paper. In practice, functional completeness may become infeasible when the number of functional models grows excessively for a given representation. Furthermore, extensions to functional models could potentially introduce cases that are geometrically infeasible due to physical constraints. Therefore, we cannot guarantee satisfiability for arbitrary functional models. However, **MSR** supports early identification of some infeasible LSMs by checking model consistency on the functional level and detecting empty sampling regions.

Future work. We plan to extend **MSR** to cover a broader set of COLREGS rules (e.g., vessels with limited maneuverability), handle vessel encounters in the presence of static obstacles, and map initial scenes into executable scenarios. For the latter, we aim to explore improved methods for trajectory generation, including dynamic trajectories and behavior modeling, as mentioned in Section 4.5.

7 Related Work

Scenario generation for ASVs. While much of the existing research on ASV safety relies on manually crafted test scenarios to find bugs in concrete ASV control software [6, 11, 54, 64], some studies have begun to explore synthetic assurance scenario generation.

Existing work dominantly relies on historical encounters [38, 62, 68]. These approaches generate assurance scenarios by extracting scenarios from real-life AIS data, followed by scenario importance evaluation. While such approaches are effective for testing the core functional correctness of an ASV control software, they are not suitable for providing assurance suites, since rare dangerous scenarios are not sufficiently covered [27].

Some studies use synthetic scenario generation approaches such as sampling-based or search-based concretization of scenes based on some set of constraints [12, 27, 49]. The authors of [49] propose an online scenario generation and selection algorithm with a discriminative neural network to assess ASV behavior in high-risk COLREGS scenarios. In [12], traffic scenarios are generated using Sobol sequences. Hazards are identified by calculating a risk vector that includes weather and traffic conditions instead of relying on COLREGS. Clustering algorithms are then used to group scenarios based on risk vectors, and the riskiest scenarios are selected to form the assurance suite. In [27], the authors use geometric constraints, search-based, and sampling-based scene concretization approaches. However, these works do not guarantee a functionally complete

assurance suite as the scenarios are generated randomly, then classified and filtered to equivalence classes as a post-processing step.

To provide robustness guarantees, test scenarios for ASVs are generated in [58] by expressing requirements in signal temporal logic and using a Gaussian process model to guide test case selection towards uncovered scenarios. In contrast, our approach uses graph solvers to systematically explore FECs to achieve functional completeness of assurance scene generation in multi-ship encounters.

Scenario generation in the automotive domain. Traffic scenario generation in the automotive domain is discussed by related work [27], where extensive research has been done in model-based safety assurance [8, 43, 57, 59, 65, 66]. In the automotive context, search-based approaches are widely employed to instantiate concrete scenarios from abstract ones [2, 10, 33, 45, 51]. For instance, [7] combines partial modeling with metaheuristic search to derive concrete scenes. However, fundamental differences in navigation rules, interaction dynamics, and environmental constraints in maritime settings prevent the direct applicability of these approaches.

Sampling-based solver frameworks such as Scenic [26, 61] are also widely used [7, 67]. Scenic employs rejection sampling for generating dynamic, stochastic scenarios and primarily supports the automotive and robotic domains. We adapt Scenic to maritime traffic in a novel way by defining condition-driven sampling. We define a multi-step refinement approach for maritime scene generation, embedding it with maritime-specific sampling conditions.

8 Conclusion

In this paper, we present **MSR** as the first end-to-end workflow for deriving functionally complete assurance suites for validating COLREGS compliance. Our approach is based on consecutive refinements from high-level functional equivalence classes (FECs) to geometrically constrained logical scenario models (LSMs). LSMs are solved using both search-based and sampling-based constraint satisfaction approaches to derive concrete initial scenes (ISs) that can be used as scenario-based evidence to demonstrate COLREGS-compliance of ASVs. We also provide a proof-of-concept demonstrator to generate COLREGS-compliant target ship trajectories from initial scenes, enabling end-to-end simulation. Our empirical evaluations show that **MSR** satisfies functional completeness, scalability, and efficiency. When compared with direct concretization (**DC**) baselines (which rely on a single LSM with a huge solution space), we find that **MSR** can reveal critical edge case scenarios that baselines miss as the number of vessels increases.

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 and is affiliated with WASP.

References

- [1] 2024. ISO/PAS 8800:2024 Road vehicles — Safety and artificial intelligence. <https://www.iso.org/standard/83303.html> Publicly Available Specification, published 13 Dec 2024.

- [2] 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 *Proc. ACM/IEEE ASE (ASE '18)*. ACM, 143–154. doi:10.1145/3238147.3238192
- [3] 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
- [4] Olov Andersson, Patrick Doherty, Mårten Lager, Jens-Olof Lindh, Linnea Persson, Elin A. Topp, Jesper Tordenlid, and Bo Wahlberg. 2021. WARA-PS: a research arena for public safety demonstrations and autonomous collaborative rescue robotics experimentation. *Auton. Intell. Syst.* 1, 1 (2021). doi:10.1007/s43684-021-00009-9
- [5] Andrea Arcuri and Lionel Briand. 2011. A practical guide for using statistical tests to assess randomized algorithms in software engineering. In *Proc. ACM/IEEE ICSE (ICSE11)*. ACM. doi:10.1145/1985793.1985795
- [6] Anna Austel, Matthias Steidel, and Bernd Westphal. 2024. Formal Specification of Traffic Scenarios in Scenario-based Testing of Maritime Assistance Systems. (2024). doi:10.5281/ZENODO.14214762
- [7] Aren A. Babikian, Oszkár Semeráth, and Dániel Varró. 2024. Concretization of Abstract Traffic Scene Specifications Using Metaheuristic Search. *IEEE Trans. Softw. Eng.* 50, 1 (2024), 48–68. doi:10.1109/tse.2023.3331254
- [8] Gerrit Bagschik, Till Menzel, and Markus Maurer. 2018. Ontology based Scene Creation for the Development of Automated Vehicles. In *Proc. IEEE IV. IEEE*, 1813–1820. doi:10.1109/ivs.2018.8500632
- [9] Earl T. Barr, Mark Harman, Phil McMin, Muzammil Shahbaz, and Shin Yoo. 2015. The Oracle Problem in Software Testing: A Survey. *IEEE Trans. Softw. Eng.* 41, 5 (2015), 507–525. doi:10.1109/tse.2014.2372785
- [10] Raja Ben Abdesslem, Shiva Nejati, Lionel C. Briand, and Thomas Stifter. 2018. Testing Vision-Based Control Systems Using Learnable Evolutionary Algorithms. In *Proc. ACM/IEEE ICSE (ICSE '18)*. ACM, 1016–1026. doi:10.1145/3180155.3180160
- [11] 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
- [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] Freddy Bouchet, Joran Rolland, and Jeroen Wouters. 2019. Rare Event Sampling Methods. *Chaos* 29, 8 (2019). doi:10.1063/1.5120509
- [14] Chong-Ju Chae. 2024. *MASS and IMO Works*. Springer Nature Switzerland, 9–27. doi:10.1007/978-3-031-69437-0_2
- [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] 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).
- [17] Indraneel Das and J. E. Dennis. 1998. Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal on Optimization* 8, 3 (1998), 631–657. doi:10.1137/s1052623496307510
- [18] Kalyanmoy Deb and Himanshu Jain. 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Trans. Evol. Comput.* 18, 4 (2014), 577–601. doi:10.1109/TEVC.2013.2281535
- [19] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* 6, 2 (2002), 182–197. doi:10.1109/4235.996017
- [20] Kalyanmoy Deb, Karthik Sindhya, and Tatsuya Okabe. 2007. Self-adaptive simulated binary crossover for real-parameter optimization. In *Proc. GECCO (GECCO '07)*. ACM, 1187–1194. doi:10.1145/1276958.1277190
- [21] T. Degré and X. Lefèvre. 1981. A Collision Avoidance System. *Journal of Navigation* 34, 2 (1981), 294–302. doi:10.1017/s0373463300021408
- [22] Renzhong Deng, Weijie Zheng, and Benjamin Doerr. 2025. The First Theoretical Approximation Guarantees for the Non-Dominated Sorting Genetic Algorithm III (NSGA-III). In *Proc. Int. Joint Conf. Artif. Intell. (IJCAI) (IJCAI-2025)*. IJCAI Organization, 8867–8875. doi:10.24963/ijcai.2025/986
- [23] 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. *J. Comput. Sci.* 67 (2023), 101937. doi:10.1016/j.jocs.2022.101937
- [24] 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
- [25] Michalis Famelis, Rick Salay, and Marsha Chechik. 2012. Partial models: towards modeling and reasoning with uncertainty (ICSE '12). IEEE Press, 573–583.

- [26] 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 *Proc. ACM SIGPLAN PLDI (PLDI '19)*. ACM, 63–78. doi:10.1145/3314221.3314633
- [27] 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 (2025), 985–1008. doi:10.1145/3728919
- [28] Benjamin Ghogh, Hadi Nekoei, Aydin Ghogh, Fakhri Karray, and Mark Crowley. 2020. Sampling Algorithms, from Survey Sampling to Monte Carlo Methods: Tutorial and Literature Review. doi:10.48550/ARXIV.2011.00901
- [29] Christoph Gladisch, Thomas Heinz, Christian Heinemann, Jens Oehlerking, Anne von Vietinghoff, and Tim Pfützer. 2019. Experience Paper: Search-Based Testing in Automated Driving Control Applications. In *Proc. ACM/IEEE ASE*. IEEE, 26–37. doi:10.1109/ase.2019.00013
- [30] 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
- [31] Baptiste Guezec, Jean-Pierre Gallois, and Frédéric Boulanger. 2024. Qualitative reasoning and cyber-physical systems: abstraction, modeling, and optimized simulation. *Innov. Syst. Softw. Eng.* 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] Fitash Ul Haq, Donghwan Shin, and Lionel Briand. 2022. Efficient online testing for DNN-enabled systems using surrogate-assisted and many-objective optimization. In *Proc. ACM/IEEE ICSE (ICSE '22)*. ACM, New York, NY, USA, 811–822. doi:10.1145/3510003.3510188
- [34] 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
- [35] Himanshu Jain and Kalyanmoy Deb. 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach. *IEEE Trans. Evol. Comput.* 18, 4 (2014), 602–622. doi:10.1109/TEVC.2013.2281534
- [36] Ulf Kargén and Dániel Varró. 2024. Towards Automated Test Scenario Generation for Assuring COLREGS Compliance of Autonomous Surface Vehicles. In *Proc. ACM/IEEE MODELS*. Vol. 222. ACM, 249–256. doi:10.1145/3640310.3674098
- [37] Lukas König, Christian Heinemann, Alberto Griggio, Michaela Klauk, Alessandro Cimatti, Franziska Henze, Stefano Tonetta, Stefan Küperkoch, Dennis Fassbender, and Michael Hanselmann. 2024. *Towards Safe Autonomous Driving: Model Checking a Behavior Planner during Development*. Springer Nature Switzerland, 44–65. doi:10.1007/978-3-031-57249-4_3
- [38] Hanna Krasowski and Matthias Althoff. 2022. CommonOcean: Composable Benchmarks for Motion Planning on Oceans. In *Proc. IEEE ITSC*. IEEE, 1676–1682. doi:10.1109/itsc55140.2022.9921925
- [39] Malik Magdon-Ismael and Amir F. Atiya. 2000. The Early Restart Algorithm. *Neural Computation* 12, 6 (2000), 1303–1312. doi:10.1162/089976600300015376
- [40] 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
- [41] 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 *Proc. IEEE/ACM Int. Conf. Softw. Eng. (ICSE), Companion Proc. (ICSE-Companion '24, Vol. 21)*. ACM, 64–68. doi:10.1145/3639478.3640045
- [42] Kristóf Marussy, Oszkár Semeráth, Aren A. Babikian, and Dániel Varró. 2020. A Specification Language for Consistent Model Generation based on Partial Models. *The Journal of Object Technology* 19, 3 (2020), 3:1. doi:10.5381/jot.2020.19.3.a12
- [43] 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 *Proc. IEEE IV*. IEEE, 2383–2390. doi:10.1109/ivs.2019.8814099
- [44] V. Lakshmi Narayanan, Ankur Gupta, Dheeraj Kumar Dhaked, Moumita Tewary, Sandeep Kumar Mondal, Ram Gopal, and Pratiksha Sarma. 2025. A systematic review on recent trends in path planning techniques for autonomous underwater vehicles. *Int. J. Intell. Robot. Appl.* 9, 3 (2025), 936–976. doi:10.1007/s41315-025-00469-9
- [45] Annibale Panichella, Fitsum Meshesha Kifetew, and Paolo Tonella. 2015. Reformulating Branch Coverage as a Many-Objective Optimization Problem. In *Proc. IEEE ICST*. IEEE, 1–10. doi:10.1109/icst.2015.7102604
- [46] 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 Trans. Softw. Eng.* 44, 2 (2018), 122–158. doi:10.1109/tse.2017.2663435
- [47] Tom Arne Pedersen, Jon Arne Glomsrud, Else-Line Ruud, Aleksander Simonsen, Jarle Sandri, 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
- [48] 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
- [49] Ivan Porres, Sepinoud Azimi, and Johan Lilius. 2020. Scenario-based Testing of a Ship Collision Avoidance System. In *Euromicro SEAA 2020*. IEEE, 545–552. doi:10.1109/seaa51224.2020.00090
- [50] Arend Rensink and Dino Distefano. 2006. Abstract Graph Transformation. *ENTCS*, 2006 157, 1 (2006), 39–59. doi:10.1016/j.entcs.2006.01.022
- [51] Vincenzo Riccio and Paolo Tonella. 2020. Model-based exploration of the frontier of behaviours for deep learning system testing. In *Proc. ESEC/FSE 2020 (ESEC/FSE '20)*. ACM, 876–888. doi:10.1145/3368089.3409730
- [52] Oszkár Semeráth, Aren A. Babikian, Anqi Li, Kristóf Marussy, and Dániel Varró. 2020. Automated generation of consistent models with structural and attribute constraints. In *Proc. ACM/IEEE MODELS (Virtual Event, Canada) (MODELS '20)*. ACM, New York, NY, USA, 187–199. doi:10.1145/3365438.3410962
- [53] Oszkár Semeráth, Rebeka Farkas, Gábor Bergmann, and Dániel Varró. 2019. Diversity of graph models and graph generators in mutation testing. *Int. J. Softw. Tools Technol. Transf.* 22, 1 (2019), 57–78. doi:10.1007/s10009-019-00530-6
- [54] Paul G. Stankiewicz and Galen E. Mullins. 2019. Improving Evaluation Methodology for Autonomous Surface Vessel COLREGS Compliance. In *Proc. IEEE OCEANS*. IEEE, 1–7. doi:10.1109/oceans.2019.8867549
- [55] Rafał 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
- [56] Trym Tengesdal, Tom Arne Pedersen, and Tor Arne Johansen. 2025. A Comparative Study of Rapidly-exploring Random Tree Algorithms Applied to Ship Trajectory Planning and Behavior Generation. *Journal of Intelligent and Robotic Systems* 111, 1 (2025). doi:10.1007/s10846-025-02222-7
- [57] Felipe Toledo, Trey Woodlief, Sebastian Elbaum, and Matthew B. Dwyer. 2024. Specifying and Monitoring Safe Driving Properties with Scene Graphs. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 15577–15584. doi:10.1109/icra57147.2024.10610973
- [58] Tobias Rye Torben, Jon Arne Glomsrud, Tom Arne Pedersen, Ingrid B Utne, and Asger J Sørensen. 2022. Automatic simulation-based testing of autonomous ships using Gaussian processes and temporal logic. *Proc. IMechE Part O* 237, 2 (2022), 293–313. doi:10.1177/1748006x211069277
- [59] S. Ulbrich, T. Menzel, A. Reschka, F. Scholdt, and M. Maurer. 2015. Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving. In *Proc. IEEE ITSC*. IEEE, 982–988. doi:10.1109/ITSC.2015.164
- [60] 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
- [61] 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
- [62] 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
- [63] 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
- [64] Kyle L. Woerner, Michael R. Benjamin, Michael Novitzky, and John J. Leonard. 2016. Collision avoidance road test for COLREGS-constrained autonomous vehicles. In *Proc. IEEE OCEANS*. IEEE, 1–6. doi:10.1109/oceans.2016.7761413
- [65] Trey Woodlief, Felipe Toledo, Matthew Dwyer, and Sebastian Elbaum. 2025. Scene Flow Specifications: Encoding and Monitoring Rich Temporal Safety Properties of Autonomous Systems. *Proc. ACM Softw. Eng.* 2, FSE (2025), 2524–2547. doi:10.1145/3729382
- [66] Trey Woodlief, Felipe Toledo, Sebastian Elbaum, and Matthew B Dwyer. 2024. S3C: Spatial Semantic Scene Coverage for Autonomous Vehicles. In *Proc. ACM/IEEE ICSE (ICSE '24)*. ACM, 1–13. doi:10.1145/3597503.3639178
- [67] Jiawei Zhang, Chejian Xu, and Bo Li. 2024. ChatScene: Knowledge-Enabled Safety-Critical Scenario Generation for Autonomous Vehicles. In *Proc. IEEE/CVF CVPR*. IEEE, 15459–15469. doi:10.1109/cvpr52733.2024.01464
- [68] Feixiang Zhu, Zhengyu Zhou, and Hongrui Lu. 2022. Randomly Testing an Autonomous Collision Avoidance System with Real-World Ship Encounter Scenario from AIS Data. *J. Mar. Sci. Eng.* 10, 11 (2022), 1588. doi:10.3390/jmse10111588
- [69] Sondre Øie, Øystein Engelhardt, and Peter Hoffmann. 2020. *Study of the Risks and Regulatory Issues of Specific Cases of MASS – Summary*. Technical Report Report No. 2020-0279, Rev. 0. DNV GL AS, Hovik, Norway.