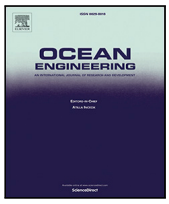
Ocean Engineering 314 (2024) 119581

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng



Research paper

Risk-aware stochastic ship routing using constrained continuous belief tree search☆

Andre Nuñez [a,b,c,*], Jennifer Wakulicz [a], Felix H. Kong [a], Alberto González-Cantos [a,d], Robert Fitch [a]

[a] The University of Technology Sydney (UTS), 81 Broadway, Ultimo, 2007, NSW, Australia
[b] Navantia Australia, 56 Pitt Street, Sydney, 2000, NSW, Australia
[c] Marintec Institute of Technology, 81 Broadway, Ultimo, 2007, NSW, Australia
[d] Navantia, Carretera de la Carraca s/n, San Fernando, 11100, Cádiz, Spain

ARTICLE INFO

Keywords:
Maritime transportation
Ship weather routing
Risk-aware planning
Voyage optimisation

ABSTRACT

Improved route planning for commercial shipping can enable reduced environmental impact, improve ship safety records, and lower fuel and maintenance costs. A fundamental challenge is to design ship routing algorithms that can contend with uncertain weather forecasts and real-world models of ship performance and safety. This paper introduces a stochastic ship routing framework that uses the conditional value-at-risk (CVaR) metric to guide the behaviour of a modified Continuous Belief Tree Search (CBTS) algorithm to find a safe and fuel-efficient long-haul shipping route. Our method provides a principled means to utilise a probabilistic representation of weather forecasts derived from ensemble forecasting for the purpose of route planning and allows for a user-defined threshold of risk tolerance. Another key advantage of our method is its ability to dynamically choose candidate route waypoints using weather-dependent estimates of fuel and safety information. Evaluation of long-haul routes through the Atlantic, Pacific and Indian oceans using real-world ship performance models and weather forecasts show considerable improvements in fuel usage and computation time compared to state-of-the-art ship routing algorithms.

1. Introduction

Transporting goods by sea supports the vast majority of global trade (Kosowska-Stamirowska, 2020). The world economy thus relies heavily on the constant operation of a large international shipping network to deliver cargo, with each ship travelling along a route through open ocean that leads to its intended destination. Choosing this route in a way that optimises certain criteria, such as safety and energy efficiency, is known as the *ship routing* problem. Ship routing is of recent interest due, in part, to the *Shipping 4.0 movement* which motivates the need for automated ship route planning (Kim et al., 2020; Montewka et al., 2018; Komianos, 2018). The promise of improved route planning is to achieve decreased fuel costs and carbon emissions by reducing fuel expenditure, and improved ship safety records by sailing through calmer seas that further reduce the economic cost and environmental impact of shipping. To do so, however, routes must be carefully planned to account for uncertain weather forecasts and also to consider trade-offs between fuel efficiency and acceptable risk of rough weather. The focus of this work is on *risk-aware ship routing* using a novel stochastic optimisation approach that addresses these trade-offs explicitly.

Existing approaches to ship routing consider the impact of weather on route safety and efficiency, but remain predominantly focused on methods that assume deterministic strategies and costs (Ksciuk et al., 2022), ignoring valuable uncertainty information. For long-haul voyages exceeding a week in duration, deterministic forecasts have low accuracy, leading to routes that incur unexpected fuel costs or unsafe weather conditions. It is therefore critical to account for weather uncertainty when planning these long-haul voyages.

Ensemble weather forecasts are the gold-standard meteorological forecast product, comprising a collection of possible future weather outcomes. None of these outcomes individually are likely to be the true future weather, but together they provide samples of likely future weather scenarios. With ensemble forecasts, one can build a probability distribution over weather conditions and, by extension, over fuel consumption or safety metrics for a given route segment. A probabilistic approach provides a way to represent the uncertainty that arises, for

☆ This project was supported by Navantia Australia and the Australian Government Research Training Program (RTP).
* Corresponding author at: The University of Technology Sydney (UTS), 81 Broadway, Ultimo, 2007, NSW, Australia.
E-mail address: andre.nunez@student.uts.edu.au (A. Nuñez).

https://doi.org/10.1016/j.oceaneng.2024.119581
Received 17 July 2024; Received in revised form 26 September 2024; Accepted 18 October 2024
Available online 4 November 2024

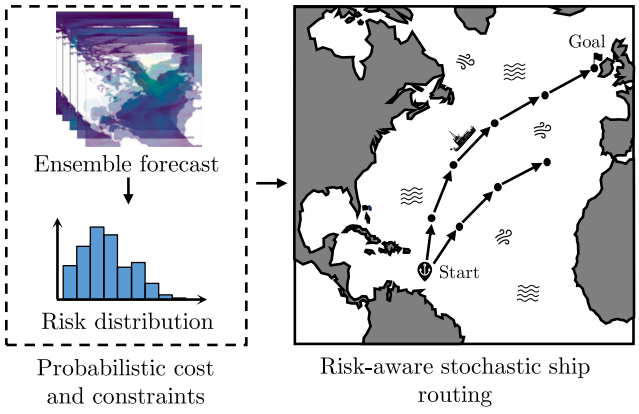


Fig. 1. An illustration of the risk-aware ship routing problem we address in this paper. Given an ensemble weather forecast, we construct probabilistic cost and constraints for planning risk-aware ship routes over continuous action space using our proposed method.

example, from conflicting forecasts in the ensemble. A major challenge remains, however, in discovering how to best use this information to build robust, efficient ship routes. Many previous studies (e.g., Mezaoui et al., 2012; Skoglund et al., 2015) simply compute the mean of fuel consumption and ship safety metrics over the ensemble. This approach often leads to overly optimistic routes that exhibit good fuel efficiency but have an unacceptably high risk of sailing through unsafe weather. The other extreme is to plan using worst-case fuel/safety estimates, leading to overly conservative routes that are safe but expend excessive fuel. The ideal metric would take into account the user's risk tolerance and thus allow for risk-aware planning.

To enable risk-aware planning under the uncertainty given by an ensemble forecast, we introduce the use of *Conditional Value-at-Risk* (CVaR), a powerful metric commonly used in finance to measure risk in investment portfolios. CVaR_α is a tunable measure of risk based on the parameter $\alpha \in [0, 1]$ that ranges continuously from the expected value of a distribution ($\alpha = 1$) to the worst-case outcome ($\alpha = 0$).

In this paper, we address the problem of risk-aware ship routing by finding a route that minimises the CVaR of the fuel consumption distribution over the voyage while keeping the CVaR of safety metric distributions below an acceptable threshold. Since CVaR can be tuned to express a user's risk appetite, this method provides flexibility to balance the trade-off between safety and fuel efficiency.

Given the user's risk appetite and an ensemble forecast, our approach uses modern path-planning methods from robotics to plan a route (see Fig. 1). We present a planning method called Constrained Continuous Belief Tree Search (C-CBTS) that dynamically constructs and searches a graph of candidate ship route waypoints to find an optimal route. Our framework modifies the continuous belief tree search (CBTS) algorithm (Morere et al., 2018), which was originally presented in the context of robotics. We adapt CBTS for ship routing by including weather dependent constraints and developing domain specific heuristics to estimate fuel consumption. Several other graph-based methods from robotics have been adapted for use in ship routing, including the Probabilistic Roadmap (PRM) algorithm (Kavraki et al., 1996; Charalambopoulos et al., 2023) and Rapidly-exploring Random Trees (RRT) (LaValle, 1998; Jang and Kim, 2022). However, these methods use only geometric information to build and search the graph, which works well for distance minimisation problems common in robots but is insufficient for ship routing which is concerned with optimising weather-dependent fuel and safety. C-CBTS is designed to optimise stochastic objective functions, such as estimated fuel use, as opposed to deterministic objectives such as geometric distance. Another advantage of CBTS is that it gains computational efficiency by using adaptive sampling to guide exploration of the search space toward promising areas.

We evaluate our method using real forecast data published by the European Centre for Medium-Range Weather Forecasts, and real-world ship performance models to inform the cost and constraint functions. On long-range routes across the Atlantic, Pacific, and Indian Oceans, our method achieves fuel savings in the range of 2.5% to 13% versus comparison methods while using 5% to 30% less computation time. These results illustrate the benefit of risk-aware planning in finding routes that are more likely to avoid storms and rough seas, and thus tend to be safer and more fuel efficient. Our results also demonstrate that our method is more computationally efficient in searching for high-quality routes due to adaptive sampling. Comparison methods search more uniformly and thus find a larger proportion of low-quality routes, only arriving at high-quality solutions after requiring a potentially untenable amount of computation time.

2. Related work

This section provides a broad overview of methods used to solve the ship routing problem and related areas of research such as robotic path planning. Broadly speaking, existing literature can be categorised into continuous or discrete approaches. We also review methods that incorporate stochasticity into their formulation.

2.1. Continuous formulations of the ship routing problem

The ship routing problem can be formulated as a continuous optimal control problem, for example (Bijlsma, 2001; Subramani et al., 2015; Subramani and Lermusiaux, 2016; Doshi et al., 2023). While these formulations use mathematical instruments with well-studied properties, it is unclear how to integrate real world ship performance and navigation data in the cost and constraint functions. This information is often drawn from lookup tables that are generated experimentally or from computational fluid dynamics studies. Ship performance curves are typically highly non-linear, non-convex and contain discontinuities, rendering them incompatible with gradient based optimisation methods that require properties such as differentiability or Lipschitz-continuity. Non-convex collision avoidance constraints arising from landmasses in navigation data also result in an unwieldy number of constraints for these methods. Consequently, the ship routing literature has primarily focused on discrete formulations of the problem that hardly rely on the mathematical properties of the objective and constraint functions.

2.2. Discrete formulations of the deterministic ship routing problem

In discrete formulations of the ship routing problem, the goal is to find the best path through a discretised representation of the state or action space. An advantage of such formulations is that it avoids solving an infinite-dimensional problem that requires finding a path in continuous space and time. Instead, these methods produce discrete paths over a finite search space.

Several distinct discrete approaches to solve the ship routing problem have been proposed. Dynamic programming based methods that search over discretised states and actions are explored in Zaccone et al. (2018) and Wei and Zhou (2012). Recently, approaches employing genetic algorithms and evolutionary computation have gained prominence (Maki et al., 2011; Yuan et al., 2022; Kuhlemann and Tierney, 2020; Zhou et al., 2023; Ma et al., 2024, 2023; Zhao et al., 2022). Similarly, reinforcement learning-based algorithms have begun to gain traction within the community, they are discussed in more detail in a recent survey paper on ship routing (Zis et al., 2020).

An important class of methods within the ship routing community are graph-based planners. The core idea uniting these methods is to construct a graph whose edges and vertices represent costs, e.g. time, carbon emissions, fuel consumption, to traverse across the ocean. Once

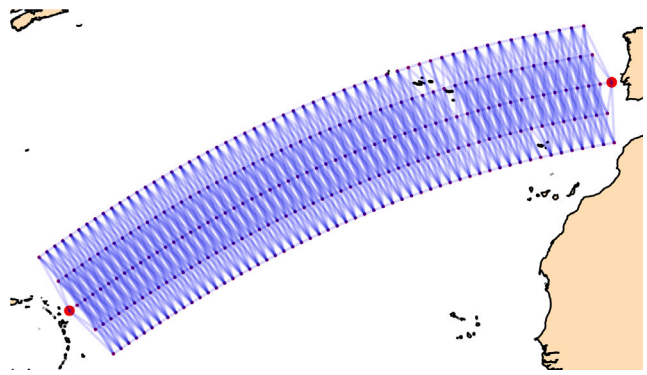


Fig. 2. An example of a heuristically generated prediscretised graph generated for a voyage between Antigua and Portugal. Red dots represent nodes and blue lines edges. Perpendiculars are formed by taking the line parallel to each point along the great circle route. Notice that the route is centred around the great circle connecting the start and end points. This approach is common in graph-based ship routing.

the graph is constructed, it is traversed using methods such as Dijkstra's algorithm to obtain an optimal path joining the start and goal locations. Examples of such approaches in the literature can be found in Wang et al. (2019), Montes (2005), Nuñez et al. (2023), Sen and Padhy (2015) and Shin et al. (2020). A crucial consideration that needs to be made is the way the graph is constructed. This heavily influences the quality of the produced solution, as the route is constrained to only use the edges that exist in the graph.

A common heuristic used to build the graph for this problem is to draw the great circle line connecting a pair of points, and casting perpendicular lines along this route at fixed distance intervals and creating nodes along these perpendiculars (Wang et al., 2018). The set of all nodes along each perpendicular are called a "stage". We refer to the ith stage along the discretised great circle path as S_i. For example, refer to Fig. 2. This method is mostly commonly justified by the fact that typically minimising time of arrival, carbon emissions or fuel consumption are the objectives of the ship routing problem. Since distance is heavily correlated with all these factors, it is conjectured that a good solution should lie near the minimum distance path, and as a result, graphs centred near it are common (Wang et al., 2019, 2018; Maki et al., 2011; Nuñez et al., 2023; Sen and Padhy, 2015).

Other classes of graph-based planners drawn from the robotic path planning community have also been proposed. Sampling-based planners build a graph by randomly sampling in the space of possible configurations and connecting these samples to form a graph joining the start and goal locations. Examples of such algorithms include Probabilistic Roadmap (PRM) and Rapidly-exploring Random Trees (RRT) among many others.

To the authors' knowledge, the only instances of sampling-based planners being applied to the ship routing problem can be found in Charalambopoulos et al. (2023) and Jang and Kim (2022). The former applies PRM to find fuel efficient routes for vessels sailing in the Mediterranean, however, no comparisons to other ship routing algorithms were made. In the latter, a combination of PRM and a modified version of RRT is used to find minimum-distance routes in the East China Sea near South Korea. Here, only distance and collision-based information were utilised in the problem formulation. More complex objectives such as fuel consumption and seakeeping constraints were not considered. These objectives require more computation to calculate the cost to travel between nodes in the graph, which can therefore lead to unwieldy runtimes.

Although algorithms that search for paths over discretised representations of state or action space often work well in practice, they suffer from the offline discretisation process that does not take the cost or constraint landscape into account. The optimal path through a graph constructed via uninformed sampling or discretisation may have much higher cost than the optimal path found in a graph constructed via informed sampling. For example, in regions where weather changes rapidly, it may be worth sampling very densely to improve the quality of the edges that are available for the route optimiser to select. This motivates the introduction of *adaptive* sampling methods for solving the ship routing problem.

Adaptive sampling methods adjust the construction of the discretised structure (e.g. graph) based on information gathered from previously added states. What makes these methods particularly appealing is that they have the benefits of other discrete methods, while also possessing the ability to add states during the planning process. Such states are typically selected because they are "informative", i.e. they improve the understanding of the planning environment, or because they "exploit", i.e., they are actions that improve the objective function.

Arguably, the most well-known algorithm in this class of methods is Monte Carlo Tree Search (MCTS) (Coulom, 2006) proposed in the artificial intelligence community. This algorithm solves decision processes with finite, discrete action spaces. The algorithm is adaptive, as it iteratively builds a search tree by using a combination of random simulations and existing information in the tree to balance exploration and exploitation to make optimal decisions in complex problems.

An extension of this method known as Continuous Belief Tree Search (CBTS) that works with continuous action spaces is presented in Morere et al. (2018). This algorithm solves an optimisation problem over a continuous estimate of the objective function using a Gaussian process (GP) associated with each node. A GP is a generalisation of a Gaussian distribution to infinite dimensions. Intuitively, it can be understood as a method to smoothly interpolate the objective function, giving an estimate of the objective function at all unobserved locations given a set of function evaluations y_i at observed locations x_i. From the observed dataset $\mathcal{D} = \{(x_i, y_i)\}$, a GP learns not only the best estimate of the objective function at any x via its mean function $\mu(x)$, but also an evaluation of the uncertainty in the estimate at x, $\sigma(x)$. Knowledge of the uncertainty of the GP's estimate at x is valuable, as it allows one to select actions that balance reducing the GP's uncertainty (thus improving its accuracy) and exploiting regions that are believed to have good objective function values.

The large amount of training data necessary to achieve an accurate estimate of complicated cost functions over large action spaces often make GP regression of the cost and constraint functions on the entire state space too cumbersome an approach. To circumvent this issue, CBTS estimates the relationship between action and cost in a local neighbourhood around samples of the full domain. Our proposed method is a novel extension of CBTS to the domain of ship routing using real-world fuel and safety constraints.

2.3. Formulations of the stochastic ship routing problem

Traditionally, ship routing methods have leaned toward deterministic approaches, e.g. Bijlsma (2001), Wang et al. (2019) and Maki et al. (2011). However, ignoring the uncertainty of the weather forecast can result in highly fuel-efficient but unacceptably dangerous routes, or overly conservative safety-conscious routes that expend significantly more fuel and carbon emissions. In contrast, meteorological forecasts commonly adopt ensemble techniques, generating a multitude of potential scenarios. Despite the availability of stochastic meteorological data, leveraging these ensemble weather forecasts for ship routing remains an under-explored area within the literature (Ksciuk et al., 2022). While some methods do incorporate ensemble forecasts, they often rely solely on the mean forecast for planning, disregarding valuable information regarding variance and uncertainty inherent in ensemble data (Lo and McCord, 1997; Yuan et al., 2022; Kuhlemann and Tierney, 2020; Yoo et al., 2021).

The value of stochastic planning has been recognised in a number of domains, for example in vehicle routing. Recent work by Kepaptsoglou

et al. (2015) has drawn inspiration from this community and introduced a chance-constrained formulation for the stochastic ship routing problem. In line with this trend, our work adopts a similar framework, utilising CVaR, which strikes a balance between expected value and worst-case, to constrain safety metrics and fuel costs. In the following section, we present the formulation of the ship routing problem that we consider.

3. Problem formulation

The problem we address is concerned with finding efficient and safe routes between a pair of locations across the ocean. Efficiency typically pertains to the energy usage or carbon footprint of the voyage. Safety involves ensuring the vessel avoids undesirable regions such as landmasses and sections of the ocean that can potentially cause harm to the ship, cargo or personnel onboard. We represent the ship's navigation through pairs of speed and heading values (also referred to as *actions* or *controls*), each sustained for a fixed period of time. Our formulation is *risk-aware* as it explicitly considers the inherent stochasticity involved in ship-routing through unpredictable weather conditions.

3.1. Ensemble weather forecasts

In order to find a risk-aware voyage, we require an accurate set of *ensemble weather forecasts*. Due to the chaotic nature of weather systems, small deviations in initial conditions and uncertainties in model parameters result in widely varying final weather predictions. One may rely on a single prediction with the highest likelihood, however, over longer horizons this is disadvantageous as forecast accuracy decreases significantly. To be more robust to the volatility in the weather, one can instead work with a set of many possible future weather outcomes, called an ensemble forecast.

Ensemble forecasting gives a more complete description of the probabilistic nature of the data, where the 'spread' or variance of outcomes reflects modelling uncertainty. This allows one to treat the ship routing problem in a more realistic framework — as a planning under uncertainty problem, rather than deterministic planning. With a full probability distribution over outcomes, ensemble forecasts enable planning of more robust routes, or even planning around lower likelihood weather outcomes that have higher expected impact on one's final plan.

To utilise statistical information provided to us by ensemble forecasts, we employ the *Conditional Value-at-Risk* (CVaR) of the efficiency and safety of the voyage generated by the ensemble. The CVaR is a single value that captures the risk profile of the ensemble forecast, allowing one to determinise the stochastic ship routing problem while still maintaining risk-awareness.

3.2. Capturing risk in ship routing via CVaR

Conditional Value-at-Risk is a risk measure developed in the finance community and has been used in a variety of domains involving problems with stochasticity. CVaR is derived by using *Value-at-Risk* (VaR), another risk measure, which is defined as the α-percentile of a random variable X, where $\alpha \in (0, 1)$.

$$\mathrm{VaR}_\alpha(X) = \min\{z | F_X(z) \geq \alpha\}, \tag{1}$$

where F_X is the cumulative distribution function of X. Plainly speaking, VaR quantifies the smallest value of a random variable for a given confidence level $1 - \alpha$. As an example, consider a hypothetical probability distribution describing the fuel cost of a fixed route under a variety of different weather conditions. If $\alpha = 0.1$, then 90% of the weather conditions along this ship route incur a cost less than $\mathrm{VaR}_{0.1}$.

CVaR is an extension of VaR. Intuitively, CVaR is the expected value of the random variable in the region defined by VaR_α. For a continuous probability distribution, CVaR is defined as (Sarykalin et al., 2008)

$$\mathrm{CVaR}_\alpha(X) = \mathbb{E}[X | X \geq \mathrm{VaR}_\alpha(X)]. \tag{2}$$

Again, if $\alpha = 0.1$, $\mathrm{CVaR}_{0.1}$ is the *expectation* of the most expensive 10% of fuel costs under the weather ensemble. If we consider the extremes, CVaR_1 is simply the expected value over the entire distribution, i.e. $\mathrm{CVaR}_1(X) = \mathbb{E}[X]$, whereas CVaR_0 is the worst case scenario over the entire distribution, i.e. $\mathrm{CVaR}_0(X) = \max X \in \Omega(X)$, where $\Omega(X)$ is the sample space of X. By varying α, it is possible to simulate a range of scenarios between the worst case and expected value.

For a discrete probability distribution, CVaR_α may be undefined for some confidence levels α. There are several definitions of CVaR for discrete probability distributions. In this paper we make use of one such definition (Sarykalin et al., 2008):

$$\mathrm{CVaR}_\alpha(X) = \lambda_\alpha(X)\mathrm{VaR}_\alpha(X) + (1 - \lambda_\alpha(X))\mathrm{CVaR}^+_\alpha(X), \tag{3}$$

where,

$$\lambda_\alpha(X) = \frac{F_X\Big(\mathrm{VaR}_\alpha(X)\Big) - \alpha}{1 - \alpha},$$

and

$$\mathrm{CVaR}^+_\alpha(X) = \mathbb{E}[X | X > \mathrm{VaR}_\alpha(X)].$$

One of the main reasons we choose to use CVaR is because it has the ability to distinguish between the severity of tail events within a region of interest. VaR, being the quantile itself, cannot.

3.3. Modelling of fuel consumption and ship safety

In order to evaluate the overall safety and efficiency of a voyage, we need an accurate model of the ship's performance subject to environmental conditions. We provide an overview of the key components of this model that will be used to calculate the objective function and constraint functions in our problem formulation. We note that in this paper, the parameters input to these models belong to a real vessel designed by Navantia Australia.

3.3.1. Fuel consumption model

The environmental conditions (wind, wave and surface ocean currents) exert forces and moments on the vessel. The resistances generated by the environment in conjunction with the desired control inputs (actions) of the vessel e.g. speed, heading, lever position are used to calculate the fuel consumption and seakeeping (safety) indices.

As described in Zaccone et al. (2018), the model used to calculate fuel consumption can be divided into the following components:

- hydrodynamic resistance calculations
- ship motion calculations
- propeller performance calculations
- engine performance calculations

The total hydrodynamic resistance of the ship, R can be expressed as

$$R = R_h + R_{aw} + R_{wind}. \tag{4}$$

The components of the resistance include the hydrodynamic resistance R_h, wind resistance R_{wind} and added wave resistance R_{aw}. Once the total resistance is calculated for a given vector of weather conditions $\mathbf{w}$ (e.g., including significant wave height, period, and direction, and others) and control inputs $\mathbf{u}$ (i.e., speed and heading) the required propeller thrust and parameters, lever position, RPM, and torque are obtained by evaluating the ship's propulsion model. The

torque and RPM are then used to determine the engine's Specific Fuel Oil Consumption (SFOC)/fuel consumption. For more detail, refer to Section 3.3 of Zaccone et al. (2018). The details of the specific implementation, particularly the coefficients of the vessel, are proprietary and will not be provided here. It is straightforward to extend this method to handle other fuel consumption models.

The fuel consumption function $F(\mathbf{u}, \mathbf{w})$, treated as a random variable, guides the minimisation of the cost function,

$$\mathrm{CVaR}_{\alpha_F}\big(F(\mathbf{u}, \mathbf{W})\big), \tag{5}$$

where $F(\mathbf{u}, \mathbf{W}) = \Big[F\big(\mathbf{u}, \mathbf{w}_1\big), \ldots, F\big(\mathbf{u}, \mathbf{w}_{N_\mathbf{W}}\big)\Big]$. Here the tensor containing the ensemble forecast is denoted as $\mathbf{W}$, and the number of members it contains as $N_\mathbf{W}$.

3.3.2. Probabilistic ship safety indices

In order to measure the safety of a voyage, we can model seakeeping responses to describe the performance of a vessel in terms of its behaviour in waves. *Seakeeping indices* are quantitative measures that have been developed using seakeeping responses to assess how well a vessel can operate in adverse weather conditions. This paper considers *slamming*, *deck wetness* and *propeller emergence*, however, the algorithm can handle any selection of seakeeping indices.

These indices can be derived by using the energy spectra of absolute ship motions (Lewis, 1988; Vettor and Guedes Soares, 2016), given by

$$S_{\eta_i} = \|\mathrm{RAO}_{\eta_i}\|^2 S_\zeta(\omega, \mu) \qquad \forall i \in \{1, \ldots, 6\}. \tag{6}$$

In Eq. (6), $\|\mathrm{RAO}_{\eta_i}\|$ represents the complex-valued response amplitude operator for each translation and rotational motion (i.e. η_i) of a ship modelled as a 3D rigid body, and μ represents the relative direction of the wave with respect to the ship. The function $S_\zeta(\omega, \mu)$ is the directional wave energy spectral density obtained by transforming the wave energy spectrum $S_\zeta(\tilde{\omega})$ through the following relation, (Lloyd, 1989)

$$S_\zeta(\omega) = S_\zeta(\tilde{\omega}) \cdot \frac{g}{g - 2\tilde{\omega} \cdot V \cdot \cos\mu},$$

where V is the speed of the ship, and g is gravitational acceleration. A wave energy spectrum is used to provide a description of the wave surface. The modified Pierson-Moskowitz wave energy spectrum is a commonly used approximation of the frequency distribution of the energy (Journée and Meijers, 1980; Lloyd, 1989):

$$S_\zeta(\tilde{\omega}) = A\tilde{\omega}^{-5} \exp\left(-B\tilde{\omega}^{-4}\right) \tag{7}$$

where

$$A = \frac{4\pi^3 H_s^2}{T_z^4}, \qquad B = \frac{16\pi^3}{T_z^4}$$

and H_s is the significant wave height and T_z is the average zero-crossing wave periods (Fossen, 2011).

Seakeeping indices are related to the moments of the energy spectra of ship motions (Eq. (6)), where the pth moment of the motion η_i is denoted as m_{p,η_i}.

The probability for the motion η_i to exceed the value γ is given by (Lloyd, 1989)

$$\exp\left(-\frac{\gamma^2}{2m_{0,\eta_i}}\right), \tag{8}$$

thus, the spectral moment of order p of wave motion η_i is (Lloyd, 1989)

$$m_{p,\eta_i} = \int_0^\infty \int_0^{2\pi} \omega^n S_{\eta_i} d\omega d\mu. \tag{9}$$

Using the moments of the energy spectra (Eq. (9)), we are now ready to introduce the seakeeping indices for propeller emergence, deck wetness and slamming (Lloyd, 1989):

$$P_1(\mathbf{u}, \mathbf{w}) := \exp\left(-\frac{D_{pe}^2}{2C_s^2 m_{0,3r}}\right) \tag{10}$$

Eq. (10) gives the probability of propeller emergence, where D_{pe} represents the effective depth of the tips of the upper propeller blades. The swell up coefficient C_s is the quotient of the actual relative motion amplitude and the notional relative motional amplitude (Lewis, 1988). The zero-th spectral moment of the notional relative motion at a given location of the ship is given by $m_{0,3r}$ and is a function of the heave and roll motions.

$$P_2(\mathbf{u}, \mathbf{w}) := \exp\left(-\frac{F_e^2}{2C_s^2 m_{0,3r}}\right) \tag{11}$$

Eq. (11) provides an expression for the deck wetness probability. The term $m_{2,3r}$ refers to the second spectral moment of the notional relative motion, measured at the appropriate location, F_e is the length of the effective freeboard (Lewis, 1988).

$$P_3(\mathbf{u}, \mathbf{w}) := \exp\left(-\frac{D_{ke}^2}{2C_s^2 m_{0,3r}} - \frac{\dot{r}_3^2}{2C_s^2 m_{2,3r}}\right) \tag{12}$$

Eq. (12) allows us to calculate the probability of slamming. Here, D_{ke} refers to the effective draft, and $\dot{r}_3^2$ can be expressed as

$$\dot{r}_3^2 = 0.093\sqrt{gL}$$

where L is the waterline length of the ship and g is gravitational acceleration (Lewis, 1988).

Rather than working with the probability of an event occurring (P_j, $\forall j \in \{1, 2, 3\}$), we are instead interested in the number of times it occurs per hour. We define C_j as the number of times the index occurs per hour.

$$C_j := \frac{3600 \cdot P_j}{\hat{T}_p} \quad \forall j \in \{1, 2, 3\}, \tag{13}$$

where $\hat{T}_p$ represents the average period of the peaks and is given by

$$\hat{T}_p = 2\pi\sqrt{\frac{m_{2,3r}}{m_{4,3r}}}.$$

Here, $m_{2,3r}, m_{4,3r}$ are the second and fourth moments of the notional relative motions.

Thus, the constraints based on each of the safety indices described in Eqs. (10), (11), (12) are given by

$$\mathrm{CVaR}_{\alpha_j}\Big(C_j(\mathbf{u}, \mathbf{W})\Big), \tag{14}$$

where $C_j(\mathbf{u}, \mathbf{W}) = \Big[C_j\big(\mathbf{u}, \mathbf{w}_1\big), \ldots, C_j\big(\mathbf{u}, \mathbf{w}_{N_\mathbf{W}}\big)\Big]$, and $j \in \{1, 2, 3\}$. With the fuel consumption model established, and the safety constraints defined by the above seakeeping indices, we are now ready to formally introduce our problem definition.

3.4. Formal problem definition: Risk-aware stochastic ship routing

Given an ensemble weather forecast and arrival and departure times, choose a sequence of *actions*, i.e. forward speed and heading, that constitute a voyage from start to goal coordinates that minimises the total CVaR_{α_F} of fuel consumption while ensuring the CVaR_{α_j} of each ship safety index C_j, $j = \{1, \ldots, N_{\text{safety}}\}$ is below a given threshold L_j for each route segment.

More formally,

Problem 1. Given start and goal locations $\mathbf{x}_{\text{start}}$ and $\mathbf{x}_{\text{goal}}$ corresponding to departure and arrival times 0 and T respectively, find a sequence of actions $\mathcal{U} = \{\mathbf{u}_0, \ldots, \mathbf{u}_{N-1}\}$ that solves

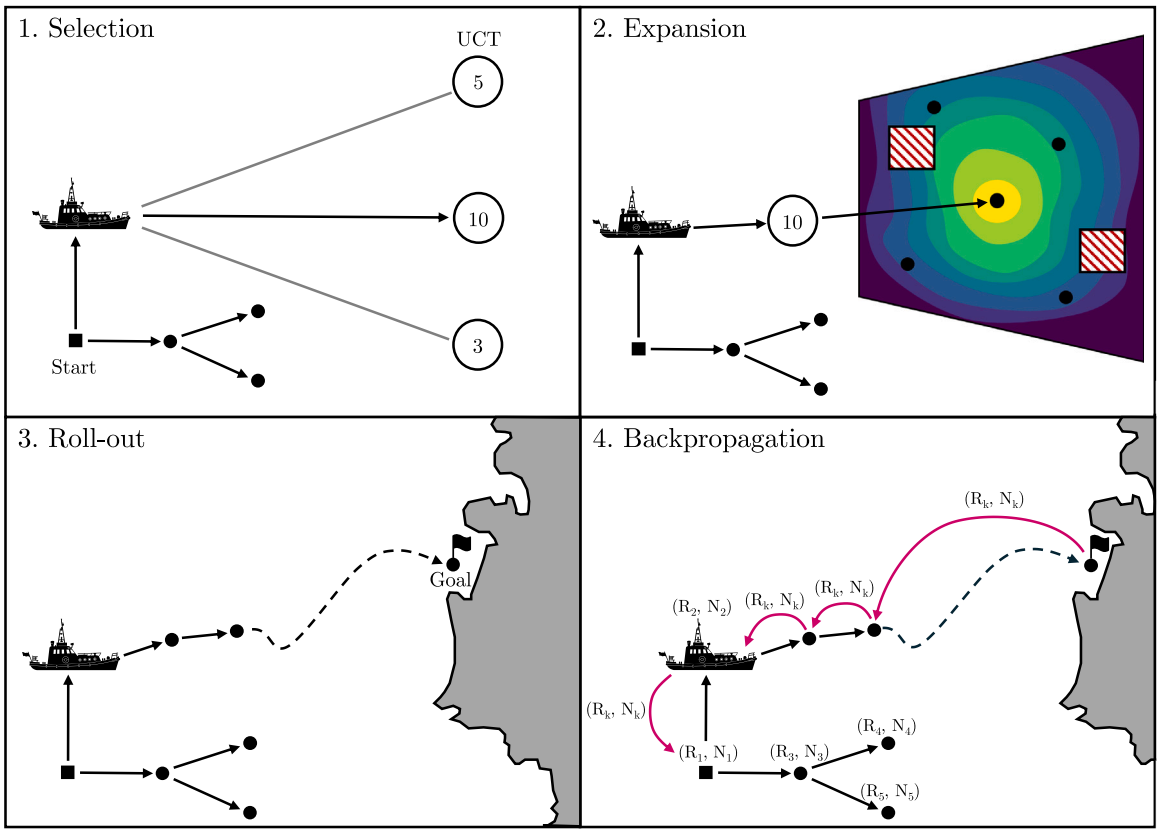


Fig. 3. An overview of constrained Continuous Belief Tree Search (C-CBTS). In each iteration of the algorithm, we *select* the branch that is most promising to explore, *expand* the tree by adding a new node. Then we simulate the remainder of the voyage via a *roll-out* and *backpropagate* the statistics of this simulation back to the root.

$$
\begin{aligned}
\min_{\mathcal{U}} \quad & \sum_{i=0}^{N-1} \mathrm{CVaR}_{\alpha_F}\left(F\left(\mathbf{u}_i, \mathbf{W}_i\right)\right), \\
\text{s.t.} \quad & \mathbf{x}_i = f(\mathbf{x}_{i-1}, \mathbf{u}_{i-1}) && \forall\, i = 1, \ldots, N \\
& \mathrm{CVaR}_{\alpha_j}\left(C_j(\mathbf{u}_i, \mathbf{W}_i)\right) \leq L_j && \forall\, i = 0, \ldots, N-1 \\
& u^- \leq u_i \leq u^+ && \forall\, i = 0, \ldots, N-1 \\
& \mathbf{u}_i \cap \mathcal{O} = \emptyset && \forall\, i = 0, \ldots, N-1 \\
& \mathrm{t}_0 = 0 \\
& \mathrm{t}_N = T \\
& \mathbf{x}_0 = \mathbf{x}_{\text{start}} \\
& \mathbf{x}_N = \mathbf{x}_{\text{goal}}
\end{aligned}
$$

where f is the ship dynamics, $\mathcal{O}$ is the set of all regions with landmasses (i.e., areas where collisions occur), $\mathbf{W}_i$ the ensemble weather forecast available at the beginning of the ith route segment, and $[u^-, u^+]$ the allowed speed and heading range of the ship. The action space for the ith segment is given by the set $[v_{\min}, \ldots, v_{\max}] \times [\Phi_{\min}, \ldots \Phi_{\max}]$, where $v_{\min}, v_{\max}$ and $\Phi_{\min}, \Phi_{\max}$ correspond to the minimum and maximum speeds and headings the vessel can take in that segment. The time at the beginning of the ith segment is denoted as t_i.

Previous work (Wang et al., 2019; Nuñez et al., 2023) reduced the computation time required for action selection in Problem 1 by discretising the action space and employing a graph-based solution. However, restricting the solution to exist only on a graph constructed without consideration of the cost landscape can result in highly sub-optimal solutions. In particular, a globally optimal solution is not guaranteed to exist within the selected discretisation. The following section introduces our approach, which is designed to overcome this limitation by adaptively selecting actions from continuous action space.

4. Constrained continuous belief tree search for risk-aware stochastic ship routing

This section presents Constrained Continuous Belief Tree Search (C-CBTS) to solve the formulation of the stochastic ship routing problem presented in Problem 1. We begin with an overview of the method before proceeding into the details. A high-level summary is illustrated in Fig. 3.

Initially, we are provided with a pair of points that we want to travel between, an ensemble weather forecast and a nautical chart detailing landmass information. Our algorithm initially creates a tree rooted at the starting point (location and time). Over time, the tree grows toward the goal point (location and time). At each iteration of the algorithm, using the weather information, we add new nodes and edges to the tree that represent segments that allow for *safe* passage, i.e. collision free and satisfy seakeeping indices, between a pair of locations. Whenever a new node and edge is added, we log the safety and fuel consumption information to keep track of the cost that is incurred. The decisions regarding new nodes to add are based on whether they improve our understanding of the viability of sailing through a particular path of the ocean or improve the overall efficiency of the voyage. The upcoming sections provide a more formal and detailed explanation of our method.

4.1. Constrained continuous belief tree search

The convention in MCTS and CBTS is to frame optimisation problems as reward maximisation. In our application, we are interested in minimising cost. These two approaches are equivalent to one another; in this paper, both "reward" and "cost" refer to fuel consumption, and we use the two interchangeably.

Constrained Continuous Belief Tree (C-CBTS) search works by growing a tree initially containing a node s representing the initial configuration (position and time) of the voyage. At each iteration, we begin

Algorithm 1 C-CBTS

Inputs: root node v_0, belief prior b_{v_0}, roll-out depth d_{max}, C-CBTS iterations N_{iter}

for $i = 1 : N_{iter}$ **do**
  $v \leftarrow$ SELECTION(v_0, d_{max})
  $v' \leftarrow$ EXPAND(v, d_{max})
  $r \leftarrow$ ROLLOUT(v', d_{max})
  BACKPROPAGATION(v', r)
return child of v_0 with maximum reward

Algorithm 2 C-CBTS selection step

function SELECTION(v_0, d_{max})
  $v \leftarrow v_0$
  while $depth(v) \leq d_{max}$ **do**
    if $|children(v)| < A_{max}$ **then**
      return v
    else
      $v \leftarrow$ maximum UCT child of v (Eqn. (15))

by making a decision regarding which node v of the tree merits further exploration. This is commonly referred to as the *selection* step, and prioritises nodes that are known to have low future cost, or have high uncertainty, which is interpreted optimistically as the *potential* for low future cost. Subsequently, the *expansion* step grows the tree toward the goal configuration g by simulating an action $\mathbf{u}_v$ from the selected node v, resulting in adding a new node v'. This action is selected using an estimate of the reward at the node v, modelled by a GP. In particular, we choose actions that yield high-reward (i.e. low cost) decision sequences further into the future (called "exploitation"), or those with highly uncertain reward, which might have better solutions (called "exploration"). Each time we choose an action, we gain new information that we use to update node v's GP. Importantly, C-CBTS extends upon the CBTS of Morere et al. (2018) by explicitly considering safety constraints, rejecting actions that violate seakeeping or result in crossing into land. Then, whenever a new node is added to the graph, we perform a *roll-out* (i.e. a simulation) that calculates an estimate of the remaining cost of the voyage from v'. This, along with the immediate cost associated with action $\mathbf{u}_v$, gives an estimate of the cost that could arise from choosing to perform $\mathbf{u}_v$ from configuration v. In order to distinguish "good" action sequences from "bad" ones, we must keep track of the average cost for each node of the tree. Therefore, after a roll-out is performed, we have new information, and so we update the averages and other relevant statistics between s and v'. This step is called the *backpropagation* step. Then, the cycle begins anew from *selection*. Refer to Algorithm 1 and Fig. 3 for a full overview of our C-CBTS implementation.

4.2. Tree branch selection

In each iteration of the algorithm, we select the branch that will be explored (see Algorithm 2). This decision is made by balancing the "exploration-exploitation trade-off". Mathematically, this is achieved by using a heuristic function called the Upper Confidence bound for Trees (UCT) to balance the exploitation of high value nodes and exploration of unvisited nodes.

For a node v, the UCT is given by

$$\mathrm{UCT}_v = \hat{\mu}_v + c\sqrt{\frac{\ln n_p}{n_v}}, \tag{15}$$

where c is an exploration parameter, n_p is the number of times that the parent node of v visited, $\hat{\mu}_v$ is the average reward of node v, and n_v the number of times that v was visited.

4.3. Tree construction and expansion with constraints

Once we have selected the branch of the tree that will be explored, we move on to the expansion step (see Algorithm 3), where we insert a new node v' that connects to the node v in the tree. This involves selecting a future action $\mathbf{u}_v$, with corresponding reward r_v, by optimising an estimate (belief) of the reward function near that node b_v, which is represented as a Gaussian process. The action-reward pair is then added to the dataset $\mathcal{D}_v = \{(\mathbf{u}_i, r_i)\}_{i=1}^{n_v}$ to train the local GP. The reward at this local GP corresponds only to the travel cost from the location of v to the point v', ignoring any reward that could be obtained after reaching v'. For an example, refer to Fig. 4. Each time the expand step visits v, a new point v' is chosen by maximising a heuristic analogous to the UCT, where exploitation of high reward actions and exploration for GP training are balanced:

$$h_v(\mathbf{u}) = \mu(b_v(\mathbf{u})) - \kappa\sigma(b_v(\mathbf{u})), \tag{16}$$

where κ is an exploration parameter, μ is the mean of the GP b_v, and σ the variance which is calculated using the covariance function k of the GP. Iterative maximisation of Eq. (16) over the action space $\mathcal{A}$ and update of b_v (lines 4–7 in Algorithm 3) amounts to a standard Bayesian optimisation (BO) procedure. As such, we refer to this step as the BO step. This attempts to reduce the number of samples required at each v in order to select a "good" action compared to random sampling. This is especially important in our problem, in which evaluating fuel consumption and seakeeping metrics are slow, due to the use of complicated real-world ship models.

To ensure that the nodes and edges added to the tree are collision free, respect seakeeping criteria and satisfy all other constraints in Problem 1 we include a rejection sampling procedure during expansion. When a new action $\mathbf{u}$ is selected in the BO step, b_v is updated with the new action-reward data. Then, the action is checked to see if it violates any seakeeping, dynamics or collision constraints (line 8, Algorithm 3). If so, the action is discarded and is not added to the search tree. An example is illustrated in Fig. 4. Due to this rejection procedure, here the cardinality of a node's children does not match the cardinality of $\mathcal{D}_v$ as it does in standard CBTS. The condition to stop expanding a node is then when it has a maximum number of children A_{max} (line 3, Algorithm 3). The user-defined A_{max} therefore gives control over the branching factor of the tree rather than the size of the training set $\mathcal{D}_v$.

Algorithm 3 C-CBTS expansion step

1: **function** EXPAND(v, d_{max})
2: $v' \leftarrow v$
3: **if** $|children(v')| < A_{max}$ **then**
4: $\mathbf{u}^* \leftarrow \arg\max_{\mathcal{A}} h_{v'}$ (Eqn. (16))
5: $v' \leftarrow$ child of v' resulting from action $\mathbf{u}^*$
6: Evaluate reward r at v'
7: Update $b_{v'}$ with new $(\mathbf{u}^*, r)$ pair
8: **if** ISINVALID(v') **then**
9: **continue**
10: **else**
11: **return** v'
12: **else**
13: **return** v'

4.4. Estimating fuel consumption and safety

After adding a new node, we refine our estimates of the fuel consumption and safety in the current branch of the tree. The first part of this process, known as roll-out, estimates the value of a visited state by simulating a sequence of actions from the current state (node) to a terminal state, e.g. the goal configuration. These actions involve visiting states that are not yet in the tree to estimate the cost of the voyage passing through a particular branch of the tree. There are many ways

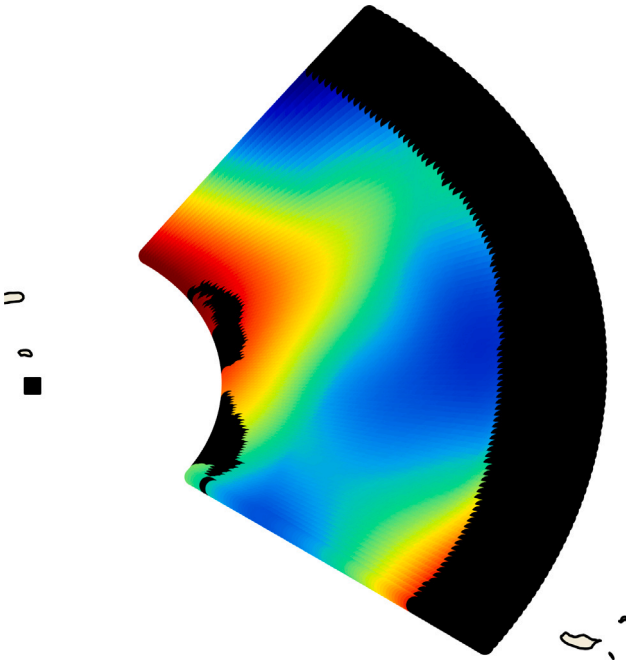


Fig. 4. Visualisation of GP reward estimate and constraints for a C-CBTS node centred around 36°30′N and 25°W at 1 December 2022 00:00:00 (black square). Actions are represented as points in the fan shaped region. Each action is a combination of a speed and heading, executed for 20 h. Black regions inside the action space indicate infeasible actions. For feasible actions, blue indicates higher reward (i.e. lower fuel cost), red indicates lower reward.

to create a function that achieves this. Section 5 details the roll-out function that we have used in our implementation.

Subsequently, we adjust the tree's predictions based on these simulations to enhance the overall accuracy of the stored estimates. In this step, known as backpropagation, the reward corresponding to this simulation is returned and stored at all nodes in the path to the visited node. A node may be selected for roll-out multiple times, and the final reward of a node v is estimated as the empirical mean $\hat{\mu}_v$ of accumulated roll-out rewards.

4.5. Route selection

Once we add the goal location to the tree and/or we exceed the computational budget is exceeded, the algorithm terminates. We can search for all the paths from the root (start) that connect to the goal using any form of graph traversal e.g. breadth first search or depth first search, and return the one which minimises the objective function, which in our case is fuel consumption. This procedure should be relatively simple, as the algorithm records the costs of all edges in the tree and only add an edge if it satisfies the constraints.

5. Experimental setup

We detail the data used for our experiments and provide an implementation description of the algorithms against which we compare our proposed method. For our C-CBTS implementation, we have created a bounding region that limits the locations of nodes in the tree. In particular, this region was created by using the polygon formed by the boundary of the prediscretised graph. An example of such a region is provided in Fig. 6. Any point outside the feasible region is discarded.

For the local GPs, we employ the Matérn 1/2 kernel with a fixed lengthscale that was hand-tuned offline. A full list of the hyperparameters for each C-CBTS experiment can be found in Appendix C. For the roll-out stage, we have developed a deterministic value function to estimate the cost to reach the goal. This function calculates the cost of the voyage from the current point to the goal using the shortest collision free path and travelling at the constant speed required to reach the goal on time.

The data sourced for this paper comes from the ECMWF's Ensemble Wave Model[1] from the period of 01-December-2022 to 15-December-2022. This model contains 50 ensembles for the entire period, where predictions are given at a $0.25° \times 0.25°$ spatial resolution and at a 6 hourly resolution. We have used data from Hybrid Ocean Coordinate Models' (HYCOM) Global Ocean Forecasting System (GOFS) model to get deterministic values for the ocean currents for the same period.

Since the underlying distribution from which the ensemble members are drawn from is not known, each member is treated as a sample from a uniform distribution, meaning they are all equally probable to occur. This decision has been made as the uniform distribution is often selected in situations where no prior information is available. It is very straightforward to adapt the proposed method if the ensemble members are weighted differently.

This work assumes that the ship's captain will adjust controls (vessel speed/heading) to ensure timely arrival at each waypoint, despite varying environmental conditions. We go to considerable lengths to verify that such controls actually exist for everyone waypoint. Our ship and weather models rigorously verify that the required speed and heading are feasible under the environmental conditions, accounting for any involuntary speed loss. Particularly, the desired speed and heading are achievable and safe (i.e. within the NORDFORSK seakeeping operability limits) across all ensemble members at the desired confidence level through appropriate configuration of the propeller, engine, and rudder.

This requires the assumption that the actual weather outcome is contained within the ensemble members. While this is still a strong assumption, it is weaker than the very common assumption in ship routing that the forecast weather matches the true weather, e.g. Zaccone et al. (2018), Wang et al. (2019), Charalambopoulos et al. (2023), Ma et al. (2023), Zhao et al. (2022), Wei and Zhou (2012), Yuan et al. (2022) and Ma et al. (2024).

Unfortunately, we are unable to disclose the vessel's parameters due to confidentiality agreements with our industry partner. However, the method developed in this study is designed to operate with black-box models, making it applicable to any vessel.

5.1. Comparison methods

Implementation details for the comparison methods employed in this paper are discussed below.

5.1.1. Prediscretised graph

In this paper, we use the 3D graph formulation presented in Wang et al. (2019) as a benchmark to compare our proposed algorithm against. For the sake of efficiency, we make use of parallel computation to pre-compute the weights of the edges in our graph in order to improve the runtime of this method. Without the advantage of parallelisation, the runtime of this method would increase by orders of magnitude, making even relatively coarse-resolution problems intractable.

In the graph, we have placed n equally spaced points along the perpendiculars, such that there are n points on either side of the great circle path connecting the start and goal. Refer to Appendix C for the n used in each experiment.

[1] The Ensemble Wave Model ©2023 European Centre for Medium-Range Weather Forecasts (ECMWF). This data is published under a Creative Commons Attribution 4.0 International (CC BY 4.0).

5.1.2. Probabilistic roadmap

We select PRM as one of the benchmarks to compare with our proposed method. In order to make this algorithm tractable for the ship routing problem, it is necessary to make some improvements. For example, Charalambopoulos et al. (2023) modify the sampling process to make the samples evenly distributed in the configuration space. Unlike typical path planning problems, where the direction of edges does not matter, for the ship routing problem, we require the edges to be directed in order to calculate the fuel consumption, seakeeping constraints and speed constraints accurately.

A modification that we have made is employing the bounding region procedure that has been used in C-CBTS to limit the locations of the nodes in the graph. Furthermore, we have developed a heuristic to define the feasible times associated with any location in the bounding region. In particular, for a given position, we use the distance to arrive at that point and the distance remaining to the goal from that point to infer a lower bound and upper bound on the speeds required to reach those points, and thus the time of arrival. The distance to arrive and distance to the goal are calculated using a Radial Basis Function interpolator using a linear kernel which is trained on data obtained from a 16-neighbour $0.25° \times 0.25°$ resolution grid.

Similar to the prediscretised graph method, we have implemented a parallelised version of PRM, where connections between the neighbours of a particular node can be computed in parallel. Without these modifications, the case studies in this paper would have been completely intractable.

5.2. Common experimental conditions

For all the experiments, we have set a hard limit of 300 s for C-CBTS to find a solution. For the other methods, we have permitted a longer runtime to accurately characterise their performance, and show that even if they are allowed to run for longer, C-CBTS finds better solutions. We also note that in terms of arrival time, we have set a 3% tolerance for arriving at the goal early/late. Moreover, since C-CBTS and PRM are randomised algorithms, we have done 30 separate runs of each (where possible) and performed an analysis of the resulting distribution of solutions. Moreover, for all the algorithms, the spatial discretisation was 5 nautical miles. This means that for each edge segment connecting a pair of points, collisions, seakeeping and cost function is aggregated in a (possibly) larger number of segments that are each 5 nautical miles long. Refer to Appendix C for a comprehensive overview of the parameters we have set for all our experiments.

6. Results

We present several case studies highlighting the advantages of risk-aware planning and efficiency of our proposed method. Section 6.1 investigates the efficacy of using CVaR in scenarios with severe weather. Subsequently, in Sections 6.2–6.4, we detail the performance of our proposed method. We consider voyages in the North Atlantic, North Pacific, and Indian Ocean, where we compare the performance of C-CBTS (the proposed algorithm) with existing graph-based methods and sampling-based planners. In these experiments, we search for fuel-efficient routes that satisfy the safety constraints, as described in our problem formulation (Section 3). For the sake of simplicity, we set $\alpha_j = \alpha_c \ \forall j \in \{1, 2, 3\}$. In other words, the safety threshold is equal to α_c that we select for all seakeeping constraints. All the voyages described in Sections 6.2–6.4 depart at 12-01-2022 00:00:00. We note that for these routes we have assigned $\alpha_f = 1$ and $\alpha_c = 1$, meaning that we have planned over the average for both the fuel consumption and seakeeping constraints. Section 6.5 investigates the impact of varying α_c in the voyages that we have considered.

Through our investigations, we have found that C-CBTS consistently performs better compared to the benchmark algorithms, both in terms of runtime and solution quality. Fuel savings of over 2.5% were consistently observed, exceeding 10% in some cases, while computation time requirements decreased by 5% or more, at times decreasing by over 60% compared to the state-of-the-art.

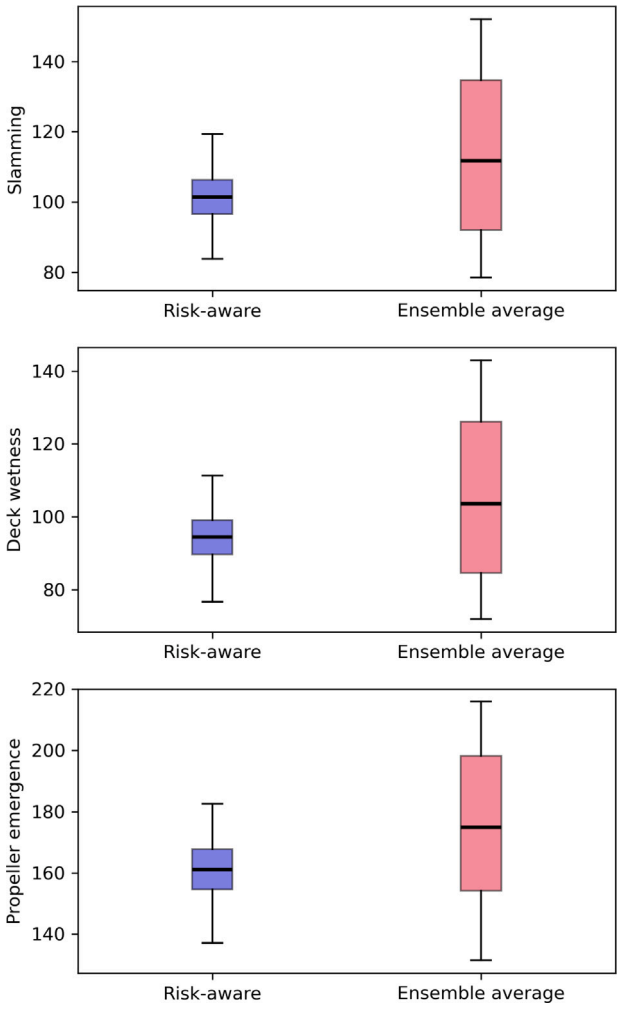


Fig. 5. A comparison of the safety indices for the $\text{CVaR}_{0.25}$ and ensemble average, routes evaluated using the control forecast (ground-truth). Each box plot represents the distribution of safety values for each constraint (slamming, deck wetness and propeller emergence) over the 30 runs. Notice that the risk-aware method consistently has a lower median and significantly smaller spread of values compared to the ensemble average.

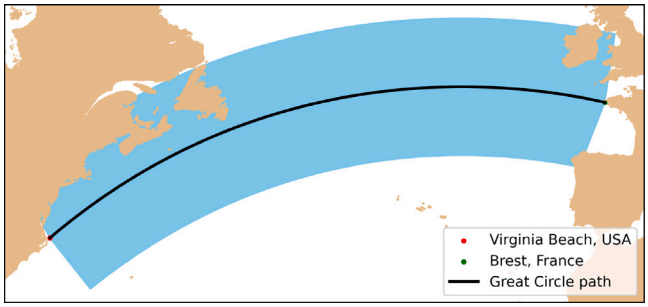


Fig. 6. A visualisation of the feasible region for an (eastward) route travelling from Virginia Beach, USA (75°4.66′W, 35°54.35′N) to Brest, France (5°W, 48°14.76′N). The (blue) region is formed by creating a polygon enclosed by the prediscretised graph, ignoring collisions, centred around the (black) line.

6.1. Case study: Advantages of risk-aware planning

We consider an eastbound voyage departing near the British Virgin Islands 20°45′51″N, 62°15′40″W travelling to Nantes, France 47°0′30″N, 2°56′29″W. The vessel departs on 1 December at 00:00:00 and arrives

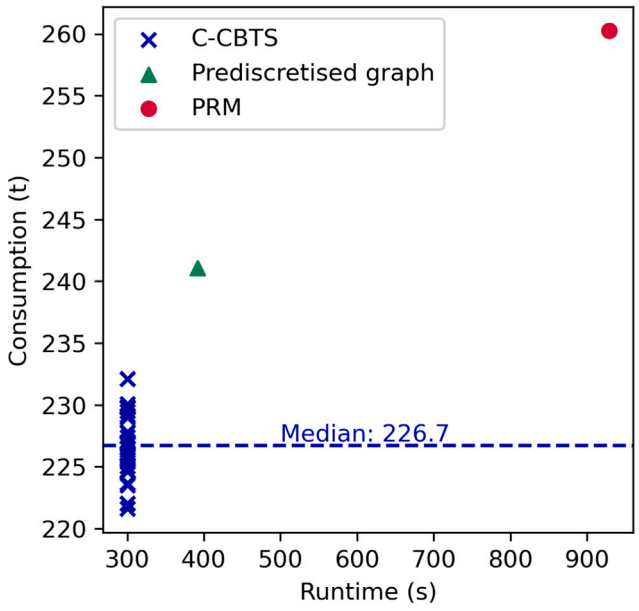


Fig. 7. Runtime (seconds) vs fuel consumption (tonnes) of the benchmarked algorithms for the transatlantic route studied in Section 6.2. The bottom left corner of the plot corresponds to solutions that are more efficient and obtained quicker. The constant speed great circle route is not shown because it was infeasible.

at Nantes at 13 December 2022 12:00:00. The distance of this voyage assuming the great circle path is taken is 3255.49 nautical miles, meaning the speed that is required to arrive on time would be 10.85 knots.

For this route, we compare the performance of risk-aware planning and "ensemble-average" planning. The latter involves planning over the average of the weather parameters across all ensemble members. We have done this comparison by running C-CBTS 30 times for each method using the same ensemble forecast data. To evaluate the performance of these routes, we use the control forecast as the ground truth. This forecast represents the best estimate of the initial state of the oceans and atmosphere, but is not considered part of the ensemble forecast; it was not made available to the route planners. For the experiments that utilised the risk-aware formulation, we have set the risk-tolerance parameter $\alpha_j = 0.25\ \forall j \in \{1,2,3\}$, i.e. we plan over the average of the worst 25% scenarios.

Our results show that the risk-aware routes are in general much safer compared to the ensemble average. In Fig. 5, we depict the distribution of values of the constraints across the 30 voyages for both methods. Notice that the risk-aware routes have both fewer occurrences of the constraint on average and a much lower standard deviation (spread). Although these routes are safer, they are slightly less energy efficient. The risk-aware routes use 248.1 tonnes of fuel on average, whereas the ensemble average routes use 245.2 tonnes, which is about a 1.1% difference. Since the risk-aware formulation is focusing on ensuring safety, it tends to avoid dangerous areas at the expense of energy efficiency. Conversely, the ensemble average route tends to be much more optimistic about the weather conditions since it takes the average, meaning it is able to plan less conservatively compared to the risk-aware formulation. However, in voyages where the weather conditions are severe, this leads to unsafe plans which may cause damage to the vessel, passengers and cargo. Even if catastrophic failure does not occur, wear and tear on the ship will accumulate more quickly using the ensemble average compared to CVaR, resulting in increased maintenance costs and downtime, reducing the productivity of the ship.

Although ensemble forecasting provides a fuller description of possible weather outcomes than a single forecast, forecasting is still imprecise for longer periods. As such, although the planner is guaranteed to generate routes that comply with safety constraints on the ensemble data, real-life weather conditions may not be accurately captured by ensemble data and safety constraints may ultimately be broken when the route is evaluated on real weather data. Despite the loss of this guarantee, our risk-aware planner still demonstrates improved safety over the ensemble average planner with reduced variance.

6.2. Case study: North Atlantic Ocean

We consider an eastward route travelling from Virginia Beach, USA (75°4.66′W, 35°54.35′N) to Brest, France (5°W, 48°14.76′N), arriving at 13 December 2022 at 12:00:00 (i.e. 300 h travelled). The length of the great circle path is 3104.85 nautical miles, meaning that assuming no weather conditions are present, the speed required to arrive on time would be 10.35 knots.

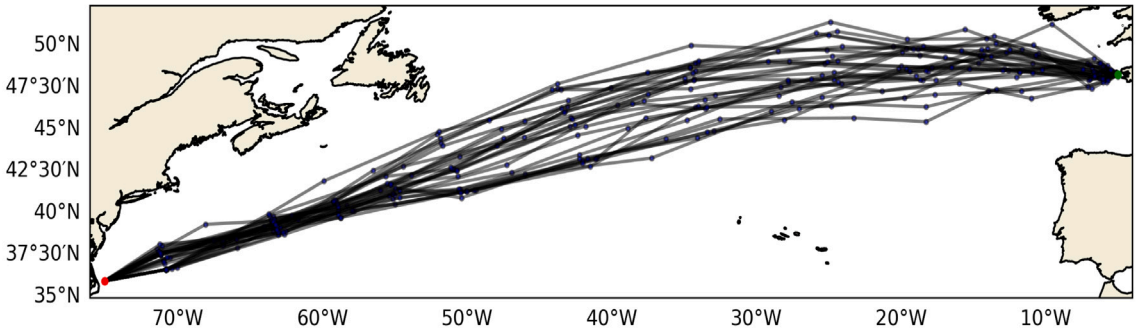
In this voyage, we found that the quality of solutions produced by C-CBTS was significantly higher than all other methods. In order to find comparable solutions, we have found that it was necessary to run the prediscretised grid and PRM for moderately to significantly longer than C-CBTS. In particular, the algorithm with the second-smallest runtime (Heuristic graph) ran for approximately 1.3 times (392 s) the amount that we allowed C-CBTS. In the worst case, PRM, ran over 3 times (952 s) the time budget allocated to C-CBTS, yet it still performed significantly worse. We note that for PRM, we have only quoted the result of a single run, which we believe is representative of the quality of the solutions produced by the algorithm for this route. This is due to the prohibitively large runtime that is required to consistently find solutions. Visualisations of the routes generated by each method are shown in Fig. 8.

The algorithm runtimes reported in this section are higher than some reported in the literature, e.g. Charalambopoulos et al. (2023); this is due to several reasons. Firstly, the use of a real vessel model as the cost function results in significantly improved accuracy at the cost of considerably higher computation. Secondly, we consider the problem of *stochastic* weather via the use of 50-member ensemble weather forecasts, increasing computation of the cost/constraint functions by a factor of 50. Despite this, the increased fidelity yields safer and more fuel efficient solutions, while requiring on the order of 5 min, which is still fast enough to run before or during a ship's voyage.

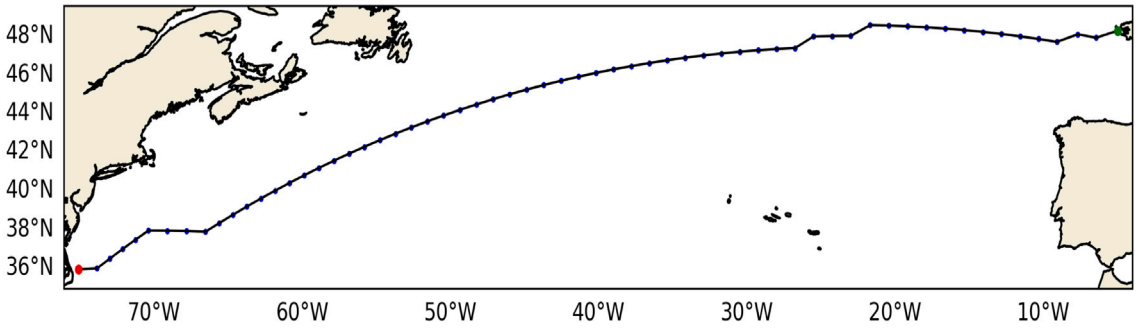
The results of each algorithm are summarised in Fig. 7, where we have plotted the runtime and solution value for each run of each algorithm we have benchmarked. Refer to Figs. 8(a)–8(c), for the paths generated by each algorithm for this experiment. The median value of C-CBTS solutions use 5.95% less fuel than the solutions found by the prediscretised graph, and 12.8% more fuel-efficient than PRM. We make note of the fact that for the prediscretised graph, a route along the great circle path was not feasible. Moreover, as can be seen, even in the worst run of C-CBTS, the quantity of fuel that is used is significantly lower than all the other algorithms. Additionally, in all 30 of the experiments that we ran, C-CBTS found a solution in all of them. We add the caveat that the success of the algorithm is dependent on tuning the hyperparameters properly such that the tradeoff between exploration and exploitation is balanced. Nevertheless, the reason C-CBTS is able to perform so well is that it is able to adaptively learn the cost function locally through a Gaussian process, whilst simultaneously being guided toward interesting regions of the search space through the "selection" step. In contrast, PRM and the prediscretised graph have no knowledge of the cost function directly, and cannot be updated "online" as the nodes and edges are predetermined.

6.3. Case study: Pacific Ocean

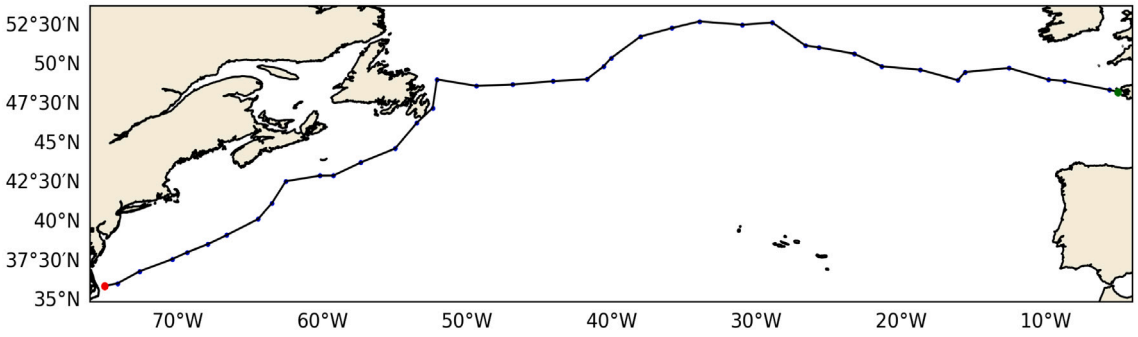
This section considers an eastward route travelling from Hawaii, USA (159°29.4′W, 22°25.5′N) to San Diego, USA (117°22.8′W, 32°38.76′N), arriving 10 December 2022 at 14:00:00 (i.e. 230 h travelled). For this voyage, the length of the great circle path for the



(a) Routes generated by C-CBTS (proposed algorithm) for the voyage presented in Section 6.2. Median consumption across routes: 226.7 t.



(b) Route generated by the prediscretised graph for the voyage presented in Section 6.2. Median consumption: 241.1 t.



(c) Route generated by PRM for the voyage presented in Section 6.2. Median consumption: 260.25 t.

Fig. 8. Visualisation of routes generated by the benchmarked algorithms for the route studied in Section 6.2, travelling from Virginia Beach, USA (75°4.66′W, 35°54.35′N) to Brest, France (5°W, 48°14.76′N), arriving at 13 December 2022 at 12:00:00 (i.e. 300 h travelled). Notice the strange shape of the PRM solution, which uses randomly sampled waypoints.

prediscretised grid is equal to 2311.63 nautical miles, meaning that a speed of 10.05 knots is required to reach the goal on time in still water.

Fig. 9 summarises the results of our experiments for this voyage, refer to Appendix A for the routes generated by our experiments. We have included the best performing great circle route for comparison. Because this voyage is shorter than the one described in Section 6.2, we were able to run PRM 30 times to create a distribution of solutions representative of its performance.

In this route, C-CBTS once again performs the best of all three benchmarked methods. However, we notice that the relative amount of fuel that C-CBTS reduces compared to the other methods reduces. In particular, with respect to the median, C-CBTS only uses 2.5% less fuel than the prediscretised grid, which again is the second best performing algorithm. Similarly, for PRM we find that compared to the previous voyage, the relative amount of fuel that is saved reduces.

We believe that the size of the search space decreasing plays a role in this result. Although, this smaller search space should benefit all benchmarks, we argue that PRM and the prediscretised grid benefit more from this compared to C-CBTS. For the prediscretised grid, having a smaller search space means that the granularity of the graph in both space and time can be increased. Similarly, for PRM, since the search space is smaller, it can cover more of it. Put another way, the resulting graph will be more dense than if compared to a region with more volume to explore. This effect of the search space being smaller is also clearly evidenced through the runtimes of these algorithms, where the times required to consistently find solutions are much closer to the 5-minute timeframe that has been allocated to C-CBTS. For C-CBTS, although the region being smaller is also beneficial, it is much less relevant since the algorithm adaptively prioritises regions which are worth exploring as opposed to exploring them uniformly as is the case for PRM and the heuristic graph.

Although overall C-CBTS was the best performing method, in this experiment there was a case where it did not produce a solution within the allotted time. Due to the probabilistic nature of the algorithm, there

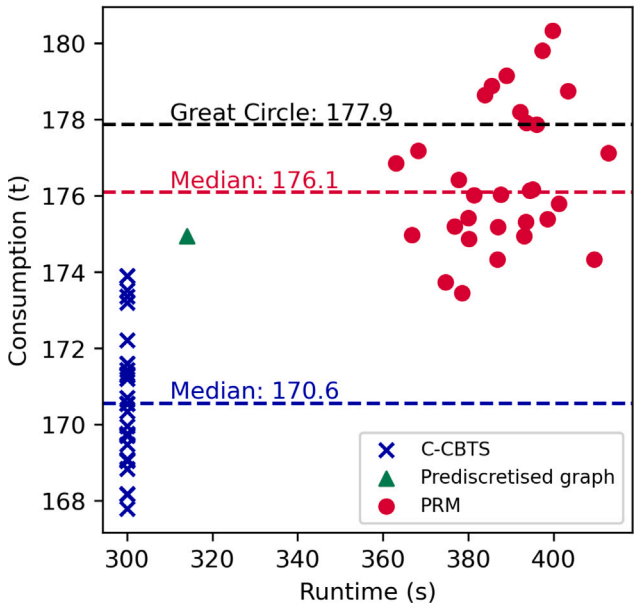


Fig. 9. Runtime (seconds) vs fuel consumption (tonnes) of the benchmarked algorithms studied in Section 6.3, the Pacific route. Points to the bottom left are better. Notice the poor performance of the great circle route.

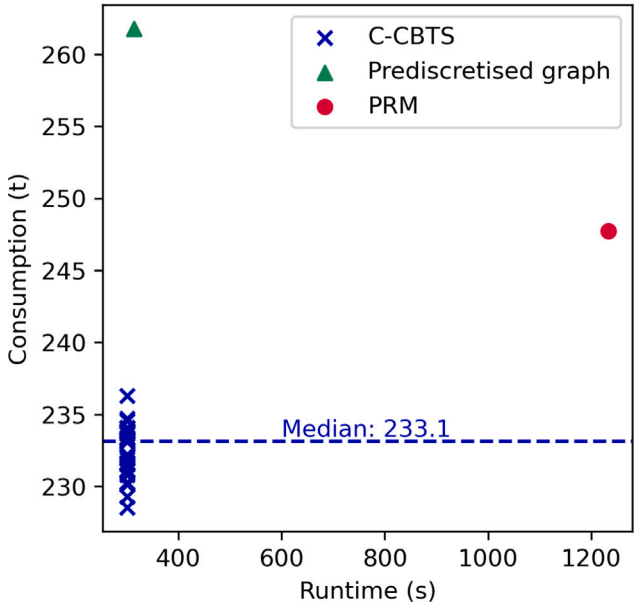


Fig. 10. Runtime (seconds) vs fuel consumption (tonnes) of the benchmarked algorithms studied in Section 6.4, the Indian Ocean route. Points to the bottom left are better. The constant speed great circle route is not shown because it was infeasible.

can be cases where it fails. It is possible to increase the likelihood of finding a solution by decreasing the hyperparameter c, which dictates the balance between exploration/exploitation. However, it should be noted that reducing this parameter will often degrade the quality of the solutions that are produced. With respect to the other methods, since the prediscretised grid is evaluated deterministically, it either always finds a solution or it does not. PRM on the other hand is not guaranteed to always find a solution, as it depends on the samples that are randomly selected. However, as mentioned before, if the search space is smaller, it will be more likely to find a path that can connect the start and goal configurations.

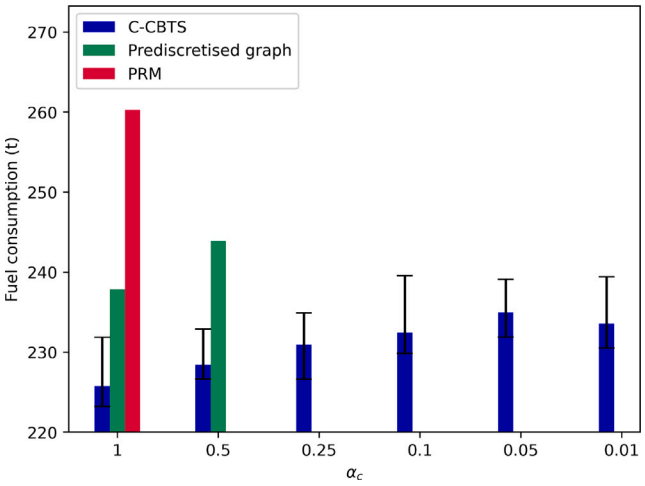


Fig. 11. Median fuel consumption across varying α_c for the transatlantic route presented in Section 6.2. Error bars represent the spread of observed values for each batch of experiments.

6.4. Case study: Indian Ocean

We consider a northward route between Perth, Australia (115°0′E, 31°58′S) to Colombo, Sri Lanka (79°37′E, 6°55′N) arriving at 13 December 2022 at 12:00:00 (i.e. 300 h travelled). The great circle path has a length of 3084.89 nautical miles, meaning that assuming no weather conditions are present, the speed required to arrive on time would be 10.28 knots.

As was the case in the North Atlantic voyage, we only report the result of a single run of PRM, due to the large runtime that is required to consistently find a solution. The results of our experiment are summarised in Fig. 10. The figures for the routes generated by this experiment are depicted in Appendix B. C-CBTS once again outperforms the rest of the benchmarks by a significant margin, where the second best performing method (PRM) uses 5.89% more fuel, and the prediscretised graph uses 10.7% more fuel.

In this route, the prediscretised graph performs quite poorly because a large detour is required to reach the goal on time to satisfy the seakeeping constraints. In particular, a route along the great circle path was not feasible. This accentuates the potential issues with using a heuristic to predetermine the nodes/edges of the graph. Of course, it is possible to amend such an issue by increasing the granularity of the graph. However, simply refining the resolution can result in intractable problems, especially for long voyages. This is also the case for PRM, where increasing the number of iterations becomes prohibitive due to the exponential increase in the number of calls to the cost function.

6.5. CVaR sensitivity analysis

The routes we have investigated in the previous sections were analysed with a fixed α_f, α_c, the CVaR risk tolerance parameter for the fuel and for safety. In this section, we demonstrate the impact of varying the risk tolerance parameter. We focus our investigation on the North Atlantic and Pacific routes. Our analysis consisted of running the experiments with varying levels of α_c for the constraint functions. The set of values for α_c considered were $[1, 0.5, 0.25, 0.1, 0.05, 0.01]$, where a batch of $n = 20$ experiments was used for each α_c, except for $\alpha_c = 1$, where $n = 30$ (results from the previous section were reused). In practice, the α_c which is selected depends on the risk tolerance of the user.

For the North Atlantic experiment, we compare C-CBTS and the prediscretised grid. PRM is excluded from this comparison due to its unwieldy runtime. In Fig. 11, we present our results. As we would

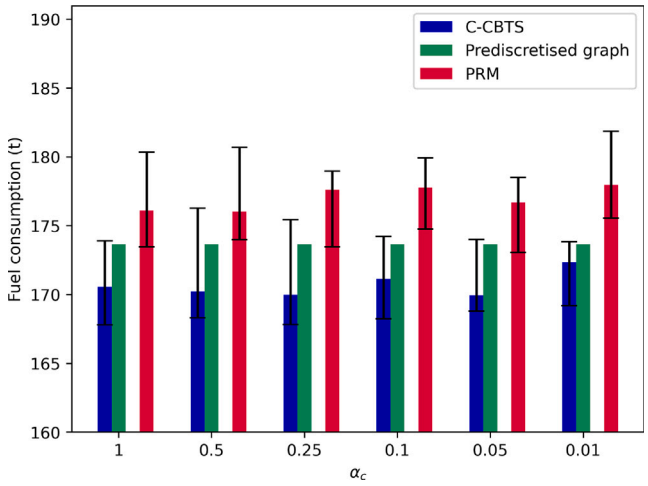


Fig. 12. Median fuel consumption across varying α_c for the Pacific route presented in Section 6.3. Error bars represent the spread of observed values for each batch of experiments.

expect, we see that overall, there is a tendency for fuel consumption to increase as we lower α_c. In scenarios where there is a large amount of variation in the weather forecasts, this is to be expected. In order to avoid dangerous regions, the ship would have to take expensive manoeuvres to avoid these zones, e.g. sail around them, pass through them quickly, or slow down and then speed up.

Observe that the prediscretised grid was only able to find a feasible solution when $\alpha_c = 0.5$, whereas C-CBTS was able to find a solution for all values of α_c. If a certain region is risky due to poor weather, if there are no edges that can allow the vessel successfully navigate through the area, it becomes impossible to find a path that connects to the goal. On the other hand, since C-CBTS builds the graph online, nodes/edges which avoid these regions can be added online. C-CBTS is able to leverage the local GPs to learn this information online at each state. As a result, it is able to consistently provide more efficient routes.

In the Pacific experiment, since the voyage and search space is smaller, we are able to include PRM. Fig. 12 shows the results of our experiments. We notice that all the algorithms were able to find a solution for all the levels of α_c specified. This suggests that the weather in this region for this period was relatively stable. In particular, the solution obtained for CVaR_1 in the prediscretised grid was identical to the one obtained in the $\text{CVaR}_{0.01}$ solution. Furthermore, unlike in the North Atlantic experiment, we observe that there is no discernible trend with respect to the amount of fuel consumed, and that the spread of the solutions across the methods is smaller.

As mentioned previously, the selection of α_c is typically defined by the user's risk tolerance. For the prediscretised graph, setting α_c too high can cause an issue of being unable to find a solution. However, for C-CBTS, this is less of a concern since it samples the search space more freely. That being said, setting an α_c that is too high can lead to routes that are unnecessarily conservative, which defeats the purpose of optimising the route.

7. Conclusion and future work

In this paper, we have introduced Constrained Continuous Belief Tree Search (C-CBTS) to the ship routing community as a method to solve a stochastic ship routing problem. We have compared the performance of the method to a number of existing graph-based methods and presented evidence to suggest that it can significantly outperform these methods over a variety of geographically diverse, long-range routes by a significant margin. Not only does C-CBTS result in more fuel-efficient routes than other algorithmic routing methods, but also requires less computation time, even after significant effort was invested in speeding up existing state-of-the-art methods. The effectiveness of C-CBTS can be attributed to the ability of C-CBTS to place nodes using fuel consumption information, rather than using only geographic information as Wang et al. (2019) and Nuñez et al. (2023) does, or doing so randomly as PRM (Charalambopoulos et al., 2023) does. C-CBTS can be used to reduce carbon emissions and fuel costs on maritime shipping routes while simultaneously improving the safety and ergonomics of crew and cargo, leading to mutual benefits to crew and to decision-makers in businesses.

The relatively generic nature of the algorithm means that it can be applied to solve any formulation of the ship routing problem, as long as it is possible to evaluate the cost and constraint functions. There are many promising directions of research that we hope can be explored in the future. For example, an extension of the algorithm that uses a dynamic control duration (dt) in order to deal with routes where the topology is more complex, for example routes that pass through narrow passages or archipelagos. A richer understanding of kernel-based methods (e.g. Gaussian processes) in the context of ship routing is another promising area of research. Finding kernels specific to ship routing would greatly improve the performance of C-CBTS and other kernel-based algorithms. These data-based methods are extremely powerful, as they can simultaneously vastly reduce the amount of compute that is required to solve the ship routing problem and improve the quality of solutions.

CRediT authorship contribution statement

Andre Nuñez: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jennifer Wakulicz:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Formal analysis, Conceptualization. **Felix H. Kong:** Writing – review & editing, Writing – original draft, Supervision, Formal analysis. **Alberto González-Cantos:** Supervision, Formal analysis, Conceptualization. **Robert Fitch:** Writing – review & editing, Writing – original draft, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Andre Nunez reports financial support was provided by Navantia SA. Andre Nunez reports a relationship with Navantia SA that includes: employment. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Pacific Ocean routes

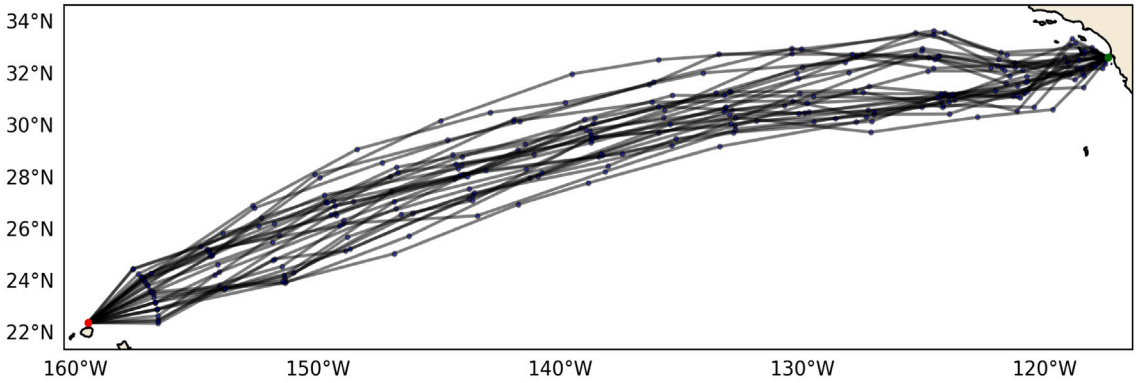
A collection of visualisations of the Pacific Ocean routes produced by the benchmarked algorithms in Section 6.3 (see Fig. 13).

Appendix B. Indian Ocean routes

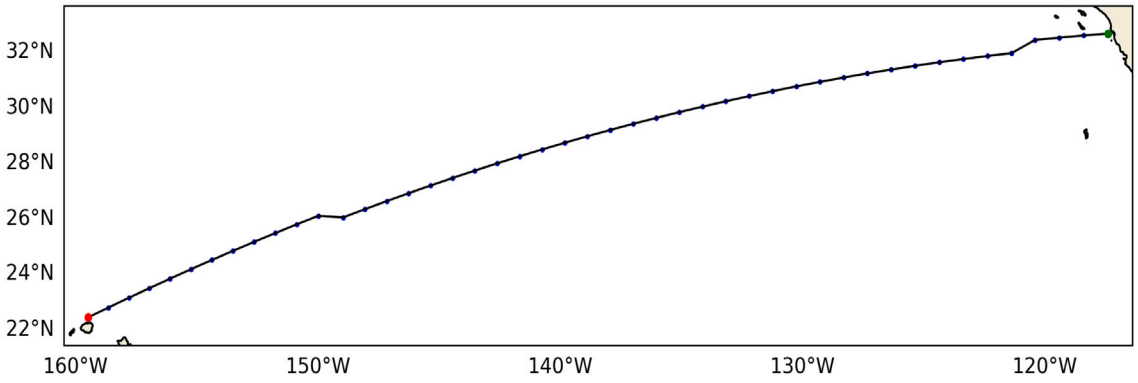
A collection of visualisations of the Indian Ocean routes produced by the benchmarked algorithms in Section 6.4 (see Figs. 14–16).

Appendix C. Experiment parameters

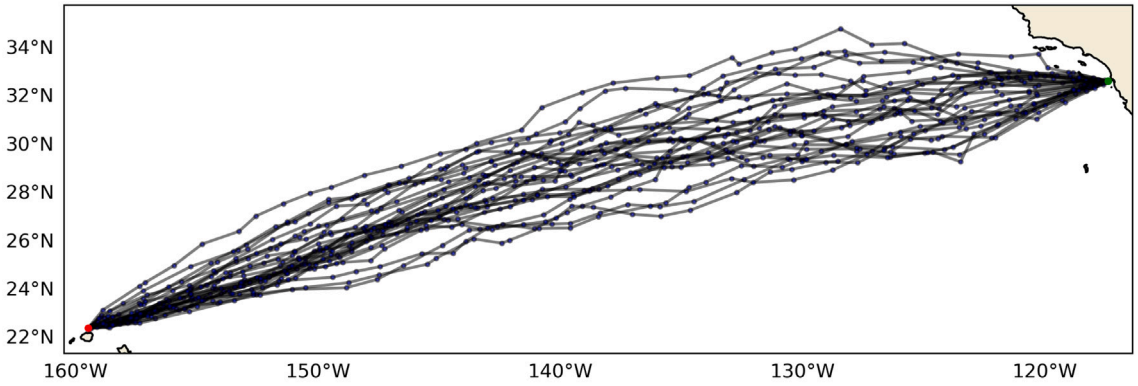
In this section, we provide a list of the parameters we have used for the different algorithms we have benchmarked for all the experiments presented.



(a) Routes generated by C-CBTS (proposed algorithm) for the voyage presented in Section 6.3. Median consumption across routes: 170.6 t.



(b) Route generated by the prediscretised graph for the voyage presented in Section 6.3. Consumption: 174.9 t.



(c) Route generated by PRM for the voyage presented in Section 6.3. Median consumption across routes: 176.1 t.

Fig. 13. Visualisation of routes generated by the benchmarked algorithms for the route studied in Section 6.2, travelling from Hawaii, USA (159°29.4′W, 22°25.5′N) to San Diego, USA (117°22.8′W, 32°38.76′N), arriving 10 December 2022 at 14:00:00 (i.e. 230 h travelled).

C.1. North Atlantic Ocean experiment

For this experiment, we have set the perpendicular length from the great circle route equal to 375 nautical miles.

For the prediscretised graph, we have set the distance between points along the perpendicular equal to 37.5 nautical miles, and the number of arrival times at each "stage", equal to 15 (details of this are explained in Wang et al. (2019)).

In our PRM implementation, we have sampled 9500 nodes, and attempted to make a connection (edge) if the node can be reached in 10 h.

We briefly describe the parameters used in C-CBTS, and then list them below.

- dt - This parameter defines the length a control action is executed for. The max depth of the tree is equal to $\frac{T}{\mathrm{dt}}$, where T is the arrival time of the vessel.
- A_{max} - This parameter is the upper bound on the number of actions that can be created for a given node in C-CBTS.
- c - This parameter controls the exploration/exploitation in Monte Carlo Tree Search, it is part of the UCT formula.
- l - This parameter is the lengthscale of the kernel of the Gaussian Process at each node in the C-CBTS tree.

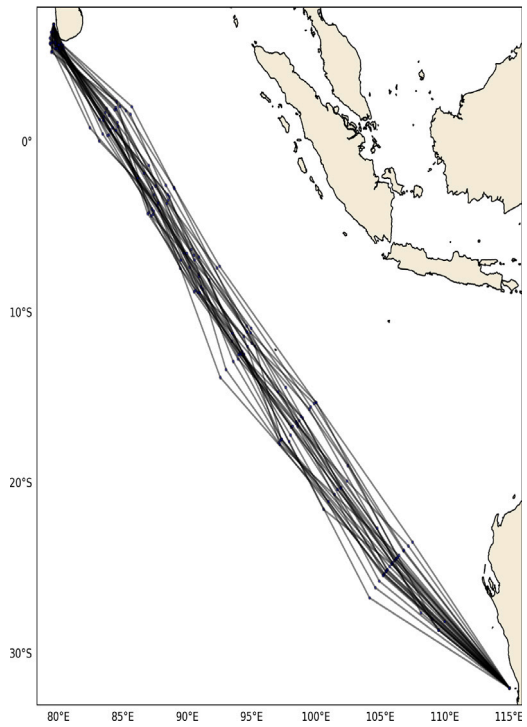


Fig. 14. Routes generated by C-CBTS (proposed algorithm) for the voyage presented in Section 6.4. Median consumption across routes: 233.1 t.

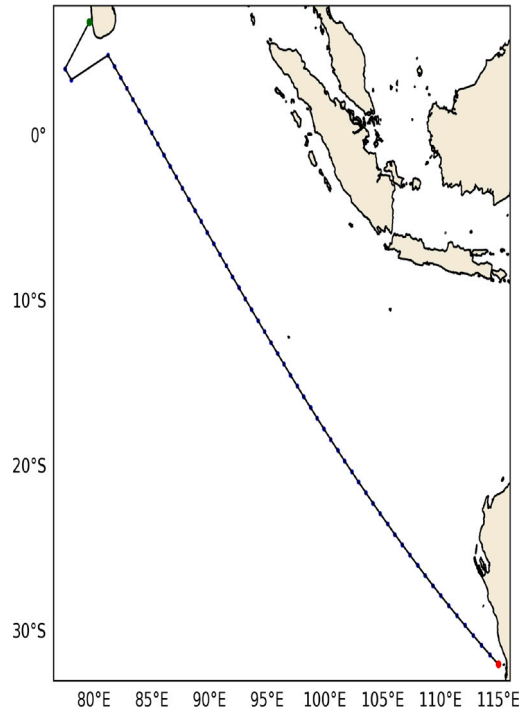


Fig. 15. Route generated by the prediscretised graph for the voyage presented in Section 6.4. Consumption: 261.8 t.

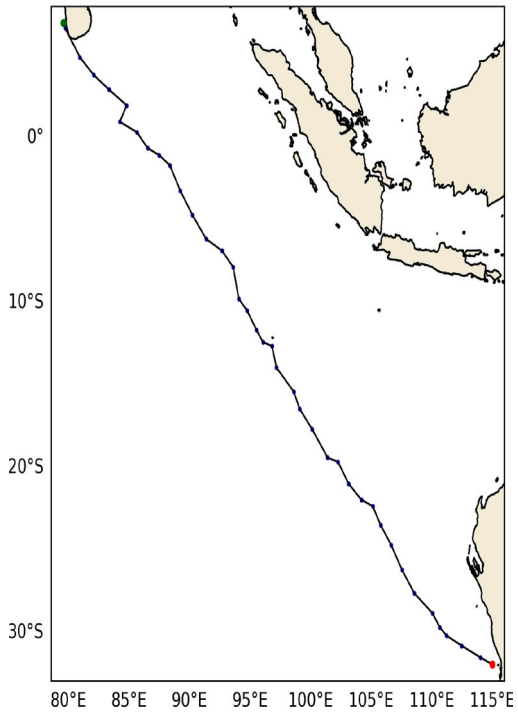


Fig. 16. Route generated by PRM for the voyage presented in Section 6.4. Consumption: 247.7 t.

- p - This parameter is the prior variance of the kernel of the Gaussian Process at each node in the C-CBTS tree. This controls the uncertainty (variance) of the Gaussian Process in the absence of data.
- κ - This parameter controls the exploration/exploitation of the Gaussian Process at each node.

The C-CBTS parameters for this experiment are given in Table 1.

C.2. Pacific Ocean experiment

For this experiment, we have set the perpendicular length from the great circle route equal to 150 nautical miles.

For the prediscretised graph, we have set the distance between points along the perpendicular equal to 25.0 nautical miles, and the number of arrival times at each "stage", equal to 20 (details of this are explained in Wang et al. (2019)).

In our PRM implementation, we have sampled 5000 nodes, and attempted to make a connection (edge) if the node can be reached in 10 h.

The C-CBTS parameters are given in Table 2.

C.3. Indian Ocean experiment

For this experiment, we have set the perpendicular length from the great circle route equal to 375 nautical miles.

For the prediscretised graph, we have set the distance between points along the perpendicular equal to 37.5 nautical miles, and the number of arrival times at each "stage", equal to 15 (details of this are explained in Wang et al. (2019)).

In our PRM implementation, we have sampled 10 000 nodes, and attempted to make a connection (edge) if the node can be reached in 10 h.

The C-CBTS parameters for this experiment are given in Table 3.

Table 1
C-CBTS parameters for Section 6.2.

Parameter	Value
dt	30
A_{max}	10
c	5.0
l	250.0
p	50.0
κ	5.0

Table 2
C-CBTS parameters for Section 6.3.

Parameter	Value
dt	23
A_{max}	10
c	5.0
l	250.0
p	50.0
κ	5.0

Table 3
C-CBTS parameters for Section 6.4.

Parameter	Value
dt	30
A_{max}	10
c	4.5
l	250.0
p	50.0
κ	5.0

References

Bijlsma, S., 2001. A computational method for the solution of optimal control problems in ship routing. Navigation 48 (3), 144–154.

Charalambopoulos, N., Xidias, E., Nearchou, A., 2023. Efficient ship weather routing using probabilistic roadmaps. Ocean Eng. 273, 114031. http://dx.doi.org/10.1016/j.oceaneng.2023.114031, URL: https://www.sciencedirect.com/science/article/pii/S0029801823004158.

Coulom, R., 2006. Efficient selectivity and backup operators in Monte-Carlo tree search. In: International Conference on Computers and Games. Springer, pp. 72–83.

Doshi, M.M., Bhabra, M.S., Lermusiaux, P.F., 2023. Energy–time optimal path planning in dynamic flows: Theory and schemes. Comput. Methods Appl. Mech. Engrg. 405, 115865. http://dx.doi.org/10.1016/j.cma.2022.115865, URL: https://www.sciencedirect.com/science/article/pii/S0045782522008210.

Fossen, T.I., 2011. Handbook of Marine Craft Hydrodynamics and Motion Control. John Wiley & Sons.

Jang, D.-u., Kim, J.-s., 2022. Development of ship route-planning algorithm based on rapidly-exploring random tree (RRT*) using designated space. J. Mar. Sci. Eng. 10 (12), 1800.

Journée, J., Meijers, J., 1980. Ship Routeing for Optimum Performance. Delft University of Technology.

Kavraki, L.E., Svestka, P., Latombe, J.-C., Overmars, M.H., 1996. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Trans. Robot. Autom. 12 (4), 566–580.

Kepaptsoglou, K., Fountas, G., Karlaftis, M.G., 2015. Weather impact on containership routing in closed seas: A chance-constraint optimization approach. Transp. Res. C 55, 139–155.

Kim, M., Joung, T.-H., Jeong, B., Park, H.-S., 2020. Autonomous shipping and its impact on regulations, technologies, and industries. J. Int. Marit. Saf. Environ. Aff. Shipp. 4 (2), 17–25.

Komianos, A., 2018. The autonomous shipping era. operational, regulatory, and quality challenges. TransNav 12 (2).

Kosowska-Stamirowska, Z., 2020. Network effects govern the evolution of maritime trade. Proc. Natl. Acad. Sci. 117 (23), 12719–12728. http://dx.doi.org/10.1073/pnas.1906670117, arXiv:https://www.pnas.org/doi/pdf/10.1073/pnas.1906670117.

Ksciuk, J., Kuhlemann, S., Tierney, K., Koberstein, A., 2022. Uncertainty in maritime ship routing and scheduling: A literature review. European J. Oper. Res. http://dx.doi.org/10.1016/j.ejor.2022.08.006.

Kuhlemann, S., Tierney, K., 2020. A genetic algorithm for finding realistic sea routes considering the weather. J. Heuristics 26 (6), 801–825.

LaValle, S., 1998. Rapidly-Exploring Random Trees: A New Tool for Path Planning. Research Report 98-11.

Lewis, E.V., 1988. Principles of naval architecture second revision. Vol. 2, Sname, Jersey, pp. 152–157.

Lloyd, A., 1989. Seakeeping: Ship Behaviour in Rough Weather. Admiralty Research Establishment, Haslar, Gosport, Publisher Ellis Horwood Ltd, John Wiley & Sons, ISBN: 0 7458 0230 3.

Lo, H.K., McCord, M.R., 1997. Strategic ship routing through stochastic ocean currents. IFAC Proc. Vol. 30 (8), 581–586. http://dx.doi.org/10.1016/S1474-6670(17)43883-3, URL: https://www.sciencedirect.com/science/article/pii/S1474667017438833, 8th IFAC/IFIP/IFORS Symposium on Transportation Systems 1997 (TS '97), Chania, Greece, 16-18 June.

Ma, W., Han, Y., Tang, H., Ma, D., Zheng, H., Zhang, Y., 2023. Ship route planning based on intelligent mapping swarm optimization. Comput. Ind. Eng. 176, 108920.

Ma, D., Zhou, S., Han, Y., Ma, W., Huang, H., 2024. Multi-objective ship weather routing method based on the improved NSGA-III algorithm. J. Ind. Inf. Integr. 38, 100570.

Maki, A., Akimoto, Y., Nagata, Y., Kobayashi, S., Kobayashi, E., Shiotani, S., Ohsawa, T., Umeda, N., 2011. A new weather-routing system that accounts for ship stability based on a real-coded genetic algorithm. J. Mar. Sci. Technol. 16, 311–322.

Mezaoui, B., Shoji, R., Tamaru, H., Nishiyama, H., 2012. A study on the use of ensemble weather forecasts for ship's weather routeing. J. Jpn. Inst. Navig. 126, 265–275.

Montes, A.A., 2005. Network Shortest Path Application for Optimum Track Ship Routing (Ph.D. thesis). Monterey, California. Naval Postgraduate School.

Montewka, J., Wróbel, K., Heikkilä, E., Valdez Banda, O., Goerlandt, F., Haugen, S., 2018. Challenges, solution proposals and research directions in safety and risk assessment of autonomous shipping. In: Probabilistic Safety Assessment and Management PSAM. Vol. 14, pp. 16–21.

Morere, P., Marchant, R., Ramos, F., 2018. Continuous state-action-observation POMDPs for trajectory planning with Bayesian optimisation. In: International Conference on Intelligent Robots and Systems. IROS, pp. 8779–8786. http://dx.doi.org/10.1109/IROS.2018.8593850.

Nuñez, A., Kong, F.H., González-Cantos, A., Fitch, R., 2023. Risk-aware stochastic ship routing using conditional value-at-risk. In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, pp. 10543–10550. http://dx.doi.org/10.1109/IROS55552.2023.10341431.

Sarykalin, S., Serraino, G., Uryasev, S., 2008. Value-at-risk vs. conditional value-at-risk in risk management and optimization. In: State-of-the-Art Decision-Making Tools in the Information-Intensive Age. Informs, pp. 270–294.

Sen, D., Padhy, C.P., 2015. An approach for development of a ship routing algorithm for application in the North Indian Ocean region. Appl. Ocean Res. 50, 173–191. http://dx.doi.org/10.1016/j.apor.2015.01.019, URL: https://www.sciencedirect.com/science/article/pii/S0141118715000206.

Shin, Y.W., Abebe, M., Noh, Y., Lee, S., Lee, I., Kim, D., Bae, J., Kim, K.C., 2020. Near-optimal weather routing by using improved A* algorithm. Appl. Sci. 10 (17), http://dx.doi.org/10.3390/app10176010, URL: https://www.mdpi.com/2076-3417/10/17/6010.

Skoglund, L., Kuttenkeuler, J., Rosén, A., Ovegård, E., 2015. A comparative study of deterministic and ensemble weather forecasts for weather routing. J. Mar. Sci. Technol. 20 (3), 429–441.

Subramani, D.N., Lermusiaux, P.F., 2016. Energy-optimal path planning by stochastic dynamically orthogonal level-set optimization. Ocean Model. 100, 57–77. http://dx.doi.org/10.1016/j.ocemod.2016.01.006, URL: https://www.sciencedirect.com/science/article/pii/S1463500316000160.

Subramani, D.N., Lolla, T., Haley, P.J., Lermusiaux, P.F., 2015. A stochastic optimization method for energy-based path planning. In: Dynamic Data-Driven Environmental Systems Science: First International Conference, DyDESS 2014, Cambridge, MA, USA, November 5-7, 2014, Revised Selected Papers. Springer, pp. 347–358.

Vettor, R., Guedes Soares, C., 2016. Development of a ship weather routing system. Ocean Eng. 123, 1–14. http://dx.doi.org/10.1016/j.oceaneng.2016.06.035, URL: https://www.sciencedirect.com/science/article/pii/S0029801816302141.

Wang, H.-B., Li, X.-G., Li, P.-F., Veremey, E.I., Sotnikova, M.V., 2018. Application of real-coded genetic algorithm in ship weather routing. J. Navig. 71 (4), 989–1010. http://dx.doi.org/10.1017/S0373463318000048.

Wang, H., Mao, W., Eriksson, L., 2019. A three-dimensional Dijkstra's algorithm for multi-objective ship voyage optimization. Ocean Eng. 186, 106131.

Wei, S., Zhou, P., 2012. Development of a 3D dynamic programming method for weather routing. TransNav 6 (1), 79–85.

Yoo, C., Lee, J.J.H., Anstee, S., Fitch, R., 2021. Path planning in uncertain ocean currents using ensemble forecasts. In: 2021 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 8323–8329.

Yuan, Q., Wang, S., Zhao, J., Hsieh, T.-H., Sun, Z., Liu, B., 2022. Uncertainty-informed ship voyage optimization approach for exploiting safety, energy saving and low carbon routes. Ocean Eng. 266, 112887.

Zaccone, R., Ottaviani, E., Figari, M., Altosole, M., 2018. Ship voyage optimization for safe and energy-efficient navigation: A dynamic programming approach. Ocean Eng. 153, 215–224.

Zhao, W., Wang, H., Geng, J., Hu, W., Zhang, Z., Zhang, G., 2022. Multi-objective weather routing algorithm for ships based on hybrid particle swarm optimization. J. Ocean Univ. China 21 (1), 28–38.

Zhou, P., Zhou, Z., Wang, Y., Wang, H., 2023. Ship weather routing based on hybrid genetic algorithm under complicated sea conditions. J. Ocean Univ. China 22 (1), 28–42.

Zis, T.P., Psaraftis, H.N., Ding, L., 2020. Ship weather routing: A taxonomy and survey. Ocean Eng. 213, 107697. http://dx.doi.org/10.1016/j.oceaneng.2020.107697, URL: https://www.sciencedirect.com/science/article/pii/S0029801820306879.