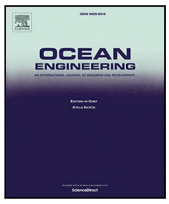
Ocean Engineering 319 (2025) 120050

Contents lists available at ScienceDirect

Ocean Engineering

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



Research paper

HADAD: Hexagonal A-Star with Differential Algorithm Designed for weather routing

Javier Jiménez de la Jara [a], Daniel Precioso [c], Louis Bu [d], M. Victoria Redondo-Neble [b], Robert Milson [d], Rafael Ballester-Ripoll [c], David Gómez-Ullate [c,*]

[a] Department of Computer Science, Universidad de Cádiz, Av. Universidad de Cádiz, 10, 11519, Puerto Real, Cádiz, Spain
[b] Department of Mathematics, Universidad de Cádiz, Av. República Árabe Saharawi, 11510, Puerto Real, Cádiz, Spain
[c] School of Science and Technology, IE University, Paseo de la Castellana, 259, 28046, Madrid, Spain
[d] Department of Mathematics and Statistics, Dalhousie University, 6297 Castine Way, B3H 3J5, Halifax, Nova Scotia, Canada

ARTICLE INFO

Keywords:
Weather routing
Decarbonization
Optimization
Seasonal study
Hexagonal grid
Variational methods

ABSTRACT

We present HADAD (Hexagonal A-Star with Differential Algorithm Designed for weather routing), a novel optimization algorithm for weather routing. HADAD conducts a global exploration using an A* search on a hexagonal grid with higher-order neighbors, enhancing directional flexibility and overcoming limitations of traditional graph searches that constrain vessel movements. It then refines the solution using a discrete Newton–Jacobi variational method, ensuring convergence to a locally optimal, smooth route in continuous space.

To evaluate the effectiveness of HADAD, we developed a benchmark comprising 1,560 instances over a full year, varying in origin–destination pairs, vessel speeds and oceanographic conditions. Our results show that HADAD outperforms pure A* graph search methods by an extra 4% savings with respect to the shortest-distance route, thanks to more flexible smoother trajectories obtained by gradient descent. In our seasonal study we observe that the savings distribution shows large seasonal variations (double savings on average in winter with respect to summer) and contains a significant number of outliers. Savings reach 27% in these cases of extreme weather events. Validation of the algorithm performed with synthetic vector fields has been conducted. In this setting, the algorithm has been adapted to handle fuel consumption optimization for Just-in-Time arrival.

By integrating global search and local optimization, HADAD effectively balances computational efficiency with route optimality, offering a practical and adaptable solution for real-world weather routing applications.

1. Introduction

Weather routing is a discipline that optimizes maritime shipping routes by leveraging weather and oceanographic data, aiming to reduce travel time, fuel consumption and emissions, or route risk, or a combination of these factors. Interest in weather routing has significantly increased in recent years driven by the shipping industry's growing focus on sustainability (IMO, 2020a; Perera and Soares, 2017). The excellent review of Zis et al. (2020) provides an extensive picture of the state of the art in weather routing up to 2020. Table 1 presents some of the most recent studies in weather routing, detailing the algorithms used and the journeys examined. Weather routing optimization algorithms can be broadly grouped into three main categories: evolutionary algorithms, graph search, and variational methods.

Evolutionary algorithms (Krata and Szlapczynska, 2018; Vettor et al., 2020; Kuhlemann and Tierney, 2020; Wang et al., 2021; Zhao et al., 2022; Grandcolas, 2022) represent the most exploratory approach, effectively identifying feasible routes in complex scenarios. However, they are inherently non-deterministic and lack guaranteed optimality (Precioso, 2023). A recent example of an evolutionary algorithm for weather routing is w-MOEA/D, developed by Szlapczynski et al. (2023). This algorithm leverages the flexibility of evolutionary methods to explore solutions that satisfy multiple constraints related to the ship's motion and safety.

Graph search techniques (Gkerekos and Lazakis, 2020; Huang et al., 2023; Zyczkowski and Szlapczynski, 2023; Chen et al., 2024) are known for their speed and flexibility (Zis et al., 2020). In this category, VISIR-2 (discoVerIng Safe and effIcient Routes), developed by Mannarini et al. (2016), Mannarini and Carelli (2019), Mannarini et al. (2024), stands out as a square-grid based graph search algorithm that dynamically updates edge weights and searches for the path with the

* Corresponding author.
E-mail addresses: javier.jimenezdelajara@uca.es (J. Jiménez de la Jara), daniel.precioso@ie.edu (D. Precioso), l.bu@dal.ca (L. Bu), victoria.redondo@uca.es (M.V. Redondo-Neble), rmilson@dal.ca (R. Milson), rballester@faculty.ie.edu (R. Ballester-Ripoll), dgomezullate@faculty.ie.edu (D. Gómez-Ullate).

https://doi.org/10.1016/j.oceaneng.2024.120050
Received 30 July 2024; Received in revised form 5 December 2024; Accepted 5 December 2024
Available online 30 December 2024
0029-8018/© 2024 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Table 1
Examples of recent weather routing studies, indicating the algorithms they used and the routes they tested on. More information about these studies can be seen in Appendix A.

Reference	Regions and Routes	Algorithm
VISIR-2 (Mannarini et al., 2024)	Mediterranean Sea (Porto Torres to Toulon, Monemvasia to Marmaris)	Graph search
Hybrid Search (Precioso et al., 2024)	North Atlantic Ocean (Charleston to Azores), Indian Ocean (Somalia to Myanmar), Caribbean Sea (Cancun to Charleston, Panama to Houston)	Evolutionary algorithm with variational refinement
PRM (Charalambopoulos et al., 2023)	Mediterranean Sea (Barcelona to Limassol, Barcelona to Thessaloniki, Barcelona to Alexandria)	Probabilistic roadmaps
OSN (Huang et al., 2023)	Pacific Ocean (Shanghai to Los Angeles, Sydney to Los Angeles), North Atlantic Ocean (New York to Rotterdam), other (Shanghai to Singapore, Singapore to Sydney, Shanghai to Sydney)	A* graph search
w-MOEA/D (Szlapczynski et al., 2023)	North Atlantic Ocean (Portugal to Azores)	Evolutionary algorithm
WRM (Grandcolas, 2022)	North Atlantic Ocean (Cap Lizard to New York)	Evolutionary algorithm
SIMROUTE (Grifoll et al., 2022)	North Atlantic Ocean (Boston to Plymouth), Mediterranean Sea (Tunis to Nice, Palma de Mallorca to Barcelona), others	A*graph search
HNDS-MPSO (Zhao et al., 2022)	Indian Ocean (Singapore to Cape Town), Pacific Ocean (Shanghai to Los Angeles)	Particle swarm
Kuhlemann and Tierney (2020)	Atlantic Ocean (New York to Paramaribo), Indian Ocean (Cape Town to Mumbai), Mediterranean Sea (Trieste to Alexandria, Algeciras to Alexandria, Rotterdam to Marseille), others	Genetic algorithm
Tsai et al. (2021)	Pacific Ocean (Taipei to Los Angeles, Tacoma to Kaohsiung)	*Non disclosed*
Wang et al. (2021)	North Atlantic Ocean (non-disclosed ports)	Genetic algorithm
Gkerekos and Lazakis (2020)	North Atlantic Ocean and Mediterranean Sea (Gulf of Guinea to Marseille)	A* graph search
Vettor et al. (2020)	Atlantic Ocean (Portugal to Azores)	Evolutionary algorithm

least CO_2 emissions. Another notable example is SIMROUTE, developed by Grifoll et al. (2022). Their A* algorithm optimizes the sailing route as a function of wave action and has been suggested as a strong candidate for comparison with other methods for weather routing.

Finally, **variational methods** (Ferraro et al., 2022; Precioso et al., 2024) excel in finding locally optimal paths, but their effectiveness is limited due to their dependence on a feasible initial solution (Precioso, 2023). In this context, we highlight the work of Ferraro et al. (2021), who developed a variational algorithm that iteratively solves a boundary value problem to quickly converge to an optimal solution for a weather routing problem based on currents or winds.

A closer analysis of Table 1 reveals a significant challenge in weather routing research: each study uses different case studies, different cost functions (some focus on minimizing travel time (Precioso et al., 2024), while others aim to reduce fuel consumption or emissions (Grifoll et al., 2022; Mannarini et al., 2024)). The lack of standardized optimization problems is a big gap in the research field of weather routing, as it has been recently noticed by Mannarini et al. (2024). When assessing the efficiency of a new methodology, researchers can only score it against basic solutions, such as the shortest-distance route (Precioso et al., 2024; Grifoll et al., 2022; Charalambopoulos et al., 2023; Grandcolas, 2022; Zhao et al., 2022). Real data are scarce and only used in some studies (Vettor et al., 2020; Gkerekos and Lazakis, 2020; Tsai et al., 2021; Szlapczynski et al., 2023). It would be desirable to establish standardized problems in weather routing research to facilitate more direct comparisons of algorithms. Other research fields where standardization of benchmarks has greatly accelerated advances are Natural Language Processing with benchmarks like MMLU (Hendrycks et al., 2021) and HellaSwag (Zellers et al., 2019), or weather prediction thanks to WeatherBench II (Rasp et al., 2024). In the last case, of particular importance to weather routing, the state of the art in short term weather prediction has recently improved thanks to ML models like Graphcast (Lam et al., 2023), whose prediction can be compared with standard models like IFS HRES (European Centre for Medium-Range Weather Forecasts, 2016) and ERA5 Forecast (Hersbach et al., 2020).

This paper introduces a new optimization method for dynamic weather routing called HADAD[1] (Hexagonal A-Star with Differential Algorithm Designed for weather routing). HADAD is in fact a pipeline of two different methods applied sequentially: first a A* graph search (Mannarini et al., 2024; Grifoll et al., 2022; Gkerekos and Lazakis, 2020; Zis et al., 2020), followed by a discrete Newton–Jacobi variational method (Precioso et al., 2024; Ferraro et al., 2022). The novelty of our A* algorithm is how the graph is constructed: we use an hexagonal grid with different resolutions, and allow the optimizer to jump over several nodes to increase the number of possible course changes. This implementation minimizes one of the major shortcomings of graph search methods: they discretize the search space and constraint the vessel movements. The second step in HADAD applied after A* refines the solution by using the A* solution as initial seed for a discrete Newton–Jacobi method that solves the discrete Euler–Lagrange equations and thus ensures that the method converges to a locally optimal solution (Ferraro et al., 2021, 2022). This second method is an adaptation of the one developed by Ferraro, Martín de Diego and Sato in the context of Lagrangian Mechanics and the Zermelo problem, and thus we will refer to it as FMS. FMS is a greedy algorithm that has been recently applied in conjunction with a shooting method for weather routing (Precioso et al., 2024) and showed great potential.

Results show that A* provides a good initial solution for FMS, but the second step in the optimization pipeline greatly improves the results since trajectories are not constrained anymore in a discrete mesh. The main motivation is building HADAD as a pipeline of two methods of a different nature is to drag the best of both approaches: A* and other graph methods are efficient in finding global optima, but the search space is constrained to trajectories that lie on a graph. FMS and

[1] Hadad, also called Ba'al-Hadad is an ancient sumerian god of rain and thunder, often depicted holding a club and thunderbolt.

other gradient descent methods are guaranteed to converge to a local optimum, but this could be far from the global one, and it strongly depends on the initial seed for the method. The combined pipeline performs a global exploration phase with A*, which is then refined by using FMS. The resulting optimal trajectory is no longer constrained to be on a graph, which allows for extra flexibility that translates into improved results (see Fig. 18).

Traditional studies often limit their focus to a single departure date per route, overlooking the inherent variability of weather conditions that require therefore adaptable and robust algorithms. Our case studies incorporate multiple departure dates for each route. This allows for the evaluation of HADAD across 52 different weeks for the same route (every week in a full year, thus enabling seasonal studies), providing comprehensive insights into our algorithm's performance and showcasing its full potential.

The paper is structured as follows. First, in Section 2, we define the specific optimization problem in weather routing that we will consider in this study: reducing travel time under a constant design speed. We then introduce the HADAD optimization algorithm in Section 3, detailing both the A* graph search used for exploration, and the Newton–Jacobi variational method used for refinement. Section 4 introduces our cases studies, which include several Origin–Destination Pairs (ODPs), departure dates over the whole year, and three different design speeds. In Section 5 we apply the HADAD optimization algorithm to these case studies. We show how the optimization results are influenced by the characteristics of every optimization problem such as the geographical location, vessel speed, the relative impact of currents and waves, and the seasons. Finally, in Section 6, we validate the usefulness of the HADAD method using synthetic vector fields to systematically evaluate its performance under controlled scenarios.

2. Optimization problem

An *instance* of a weather routing problem should include the following components: origin and destination, departure date and time, vessel characteristics, consumption model (or more generally cost function), relevant meteorological information (e.g., waves, currents) and decision variables available to the algorithm to minimize the cost function. For this study, the choices for each of the above elements are:

- The **cost function** is the total travel time from the origin to the destination. We assume that the vessel's engine power remains constant along the route, resulting in a constant design speed in calm water (SCW). The weather conditions (currents and waves) cause involuntary speed reductions, which in turn affect the total travel time. The specific model is discussed in Section 2 below.
- The **decision variables** for the optimization problem are the vessel's waypoints between origin and destination, or equivalently the heading of the vessel at certain time intervals. A solution to the optimization problem is given as a polygonal curve defined by a sequence of waypoints $\{\mathbf{q}_i\}_{i=1}^{M}$, where $\mathbf{q}_i = [u_i, v_i, t_i]$, and M is the total number of waypoints along a route. Here, $\mathbf{q}_1$ is the origin and $\mathbf{q}_M$ is the destination. Each waypoint is a triple, where each entry represents the longitude (u), latitude (v), and timestamp (t) at the ith waypoint, respectively. For simplicity, we will refer to the spatial coordinates as $\mathbf{x} = [u, v]$. The decision variables for the optimization problem are thus to determine the vessel's waypoints between origin (point $\mathbf{x}_1$) and destination (point $\mathbf{x}_M$), $\{\mathbf{q}_i\}_{i=2}^{M-1}$.
- The **origin and destination points** can be placed anywhere on the ocean. We have chosen several representative origin–destination pairs (ODPs) to account for geographical variation. This choice is shown in Table 4.
- **Departure dates** for this study range from January 1st 2023 to December 31st 2023, considering a departure every Sunday at 00:00 CET. These 52 different dates will allow to explore seasonal effects.
- **Vessel characteristics** relevant to our cost function include the beam length (L), displacement (∇) and block coefficient (c_B), as discussed in Section 2.1.1. In addition, the design speed in calm water (SCW) is also fixed throughout the route.
- The **meteorological data** used in this study includes non-stationary ocean currents and waves. These data come from standard weather and ocean forecasts from Copernicus CMEMS and NOAA. Full details are given in Section 4.1 and Appendix B.

The emphasis of this paper is on the novelty of the HADAD algorithm explained in Section 3, rather than achieving a complex consumption model. HADAD can be adapted with ease to any consumption model. Although the results in Section 5 refer to the choices specified above, it is worth mentioning that HADAD can also be adapted to handle other cost functions, such as fuel optimization at a prescribed trip duration (relevant to Just-in-Time arrival). This is explained in Section 6, together with a numerical validation of the algorithm on standard synthetic benchmarks.

2.1. Cost function

The cost function in this study is the travel time. In the remainder of this section, we explain how to calculate the travel time based on the speed over ground, accounting for the effects of waves and ocean currents.

The vessel's Speed Through Water (STW) is the speed relative to the water after accounting for involuntary speed reductions due to waves. This speed is represented by the vector **STW**, which has a heading angle ψ_{tw} (measured in degrees relative to true north). The STW vector is influenced by weather conditions and the vessel characteristics. Consequently, the ship's Speed Over Ground (SOG) represents the actual movement of the vessel over the ground after accounting for all external factors, with its course over ground angle denoted as ψ_{og}. With this in mind, our cost function, the total travel time can be computed as a function of the distance traveled and the SOG, which depends on both the navigation conditions and the selected route.

2.1.1. Effect of waves

First, we analyze the impact of wave-induced speed reduction. Since the 1970s, both academia and industry have extensively studied this effect. Various methods have been developed, ranging from statistical and regression models based on wave tank experiments (Faltinsen, 1980; Holtrop et al., 1982; Townsin and Kwon, 1983; Young-Joong, 2005; Lang and Mao, 2020) to advanced fluid dynamics and potential flow models (Kim et al., 2017). Recently, data-driven methods, including machine learning and neural networks (Kim and Steen, 2022; Lang et al., 2022; Mittendorf et al., 2022; Yang et al., 2024), have further improved our ability to predict wave resistance and the resulting speed losses.

We have chosen the Townsin–Kwon model (Townsin and Kwon, 1983) as presented in Molland et al. (2011), as it provides valid results for a range of environmental parameters: significant wave height H_{s} (in meters) and wave incidence angle δ_{s} (in degrees) relative to the ship's heading through water (ψ_{tw}). It also depends on the design speed in calm water SCW (in meters per second), the vessel's length L (in meters), displacement ∇ (in cubic meters), and block coefficient c_B.

According to Young-Joong (2005) and Molland et al. (2011), the vessel's Speed Through Water under wave conditions is given by:

$$\text{STW} = c_{\text{s}} \cdot \text{SCW}, \tag{1}$$

where c_{s} is the speed reduction coefficient due to waves, which is decomposed into three factors:

$$c_{\text{s}} = c_\beta \cdot c_u \cdot \alpha, \tag{2}$$

where c_β is the weather reduction coefficient, c_u is the speed reduction coefficient related to the vessel's displacement, and α is the correction factor based on the vessel's Froude number.

The weather reduction coefficient c_β depends on the wave incidence angle δ_s and the Beaufort number BN:

$$c_\beta := \begin{cases} 1 & 0° \le \delta_s < 30° \\ 0.85 - 0.015 \cdot (\mathrm{BN} - 4)^2 & 30° \le \delta_s < 60° \\ 0.45 - 0.030 \cdot (\mathrm{BN} - 6)^2 & 60° \le \delta_s < 150° \\ 0.20 - 0.015 \cdot (\mathrm{BN} - 8)^2 & 150° \le \delta_s \le 180° \end{cases} \tag{3}$$

The wave incidence angle δ_s is defined as:

$$\delta_s = |\psi_{tw} - \psi_s|, \tag{4}$$

where ψ_{tw} is the vessel's heading through water and ψ_s is the wave's direction, both measured relative to true north.

The Beaufort number BN is a dimensionless quantity usually derived from wind speed (National Oceanic and Atmospheric Administration (NOAA), a), but it can also be related to H_s using the relation between wind speed and wave formation (National Oceanic and Atmospheric Administration (NOAA), b):

$$\mathrm{BN} = \left(\kappa H_s\right)^{2/3}; \quad \kappa = 2.68\ \mathrm{m}^{-1}. \tag{5}$$

The speed reduction coefficient c_u depends on the Beaufort number BN and the displacement of the vessel ∇ (in cubic meters):

$$c_u = 0.7\ \mathrm{BN} + \frac{\mathrm{BN}^{13/2}}{\theta \nabla^{2/3}}; \quad \theta = 22\ \mathrm{m}^{-2}. \tag{6}$$

The correction factor α depends on the vessel's Froude number Fr and its block coefficient c_B. For our simulation, we use the formula from Molland et al. (2011) for a block coefficient $c_B = 0.6$:

$$\alpha = 2.2 - 2.5\ \mathrm{Fr} - 9.7\ \mathrm{Fr}^2, \tag{7}$$

where the Froude number Fr is defined as:

$$\mathrm{Fr} = \frac{\mathrm{SCW}}{\sqrt{gL}}, \tag{8}$$

with SCW being the vessel's design speed in calm water, L the length of the vessel, and g the acceleration due to gravity.

Note that we consider only the *involuntary speed reduction*, which occurs due to the mechanical effect of added resistance from waves. The *voluntary speed reduction*, resulting from the captain's decision to lower the thrust for safety or efficiency, is not included in this model.

A requirement of the optimization algorithm we present in this paper (Section 3.2) is to evaluate first- and second-order derivatives of the cost function with respect to the decision variables. The coefficient c_β in Eq. (3) is a piecewise function of the wave incidence angle δ_s, and this lack of continuity can cause numerical instability in the optimization algorithm. To address this, we have substituted c_β with a smooth function, as shown in Fig. 1. The smooth $\hat{c}_\beta$ is given by:

$$\hat{c}_\beta = c - b \cdot (\mathrm{BN} - a)^2, \tag{9}$$

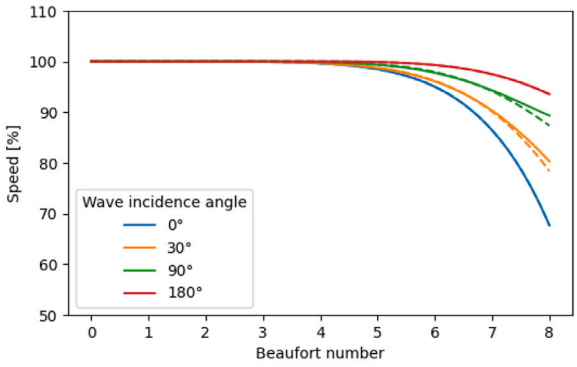
where:

$$\begin{aligned} a &= 6 \sin^{(2/3)}\left(\delta_s/2\right) + 2, \\ b &= \frac{1}{80}\left[1 + \sin\left(1.2\ \delta_s\right) - \cos\left(1.2\ \delta_s\right)\right], \\ c &= 1 - 0.8 \sin\left(\delta_s/2\right). \end{aligned} \tag{10}$$

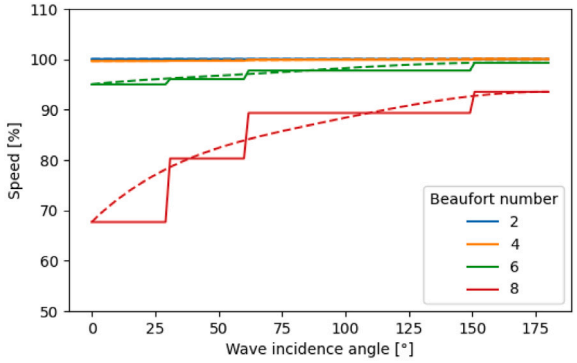
2.1.2. Effect of ocean currents

The other major component affecting vessel speed is ocean currents. Unlike waves, which only reduce the vessel's speed magnitude, ocean currents $\mathbf{w}$ change both the magnitude and direction of the vessel's motion over ground. The vessel's Speed Over Ground (SOG) is obtained by vector addition of the Speed Through Water (STW) and the current vector:

$$\mathbf{SOG} = \mathbf{STW} + \mathbf{w} \tag{11}$$



(a) Speed loss as a function of the Beaufort number, at different wave incidence angles. The smooth function closely approximates the original while eliminating discontinuities.



(b) Speed loss as a function of the wave incidence angle, at different Beaufort numbers. The smoothing effect improves the simulation by eliminating abrupt transitions.

Fig. 1. Comparison of $\hat{c}_\beta$ (dotted line) defined in Eq. (9) versus the non-smooth c_β from Molland et al. (2011) (solid line).

Here, **STW** is the vessel's speed through water vector, aligned with the heading ψ_{tw}, and $\mathbf{w}$ is the current vector with magnitude w and direction ψ_c (relative to true north).

To maintain a desired course over ground ψ_{og}, the vessel adjusts its heading ψ_{tw} to compensate for the effect of currents. The relationship between the heading and the course over ground is given by:

$$\sin(\psi_{tw} - \psi_{og}) = -\frac{w}{\mathrm{STW}} \sin(\psi_c - \psi_{og}) \tag{12}$$

This equation ensures that the cross-track component of the current is counteracted by adjusting the heading, allowing the vessel to stay on the desired course over ground. The magnitude of the Speed Over Ground is calculated as:

$$\mathrm{SOG} = w \cos(\psi_c - \psi_{og}) + \sqrt{\mathrm{STW}^2 - w^2 \sin^2(\psi_c - \psi_{og})} \tag{13}$$

The travel time δt between two waypoints separated by a distance δx is then:

$$\delta t = \frac{\delta x}{\mathrm{SOG}} \tag{14}$$

It is important to ensure that the vessel's speed through water is sufficient to overcome the perpendicular component of the current. This is feasible if:

$$\frac{w}{\mathrm{STW}} \le 1 \tag{15}$$

This condition ensures that the sine function in Eq. (12) remains within its valid range.

Since the environmental fields that influence SOG vary with both space and time, the value of δt is computed using interpolation procedures, as detailed in Section 3.1.3.

This completes the definition of the optimization problem to be tackled in this study. Future versions of the HADAD algorithm will include other cost functions, such as fuel consumption, greenhouse gas emissions, or safety at sea, and will consider additional decision variables, such as the ability to change engine power along the route.

2.1.3. Fuel consumption and just-in-time arrivals

According to the traditional Holtrop model (Holtrop et al., 1982), fuel consumption is primarily dependent on the vessel's engine power and the voyage duration. Under the assumption of constant engine power, minimizing travel time is directly related to minimizing fuel consumption. However, following the Just-In-Time (JIT) guidance from the International Maritime Organization (IMO) (IMO, 2020b), a more sophisticated optimization approach would focus on minimizing fuel consumption while adhering to a prescribed maximum travel duration. The JIT guidance emphasizes operational efficiency by aligning vessel operations with a fixed time of arrival rather than simply reducing travel time. Future versions of our HADAD optimization algorithm will incorporate these principles by allowing for adjustments in both course over ground and engine power along the route, and will potentially include safety parameters. For this initial version, the focus on minimizing travel time with constant engine power provides a solid foundation for route planning and establishes a valid starting point for future development.

3. HADAD algorithm

HADAD algorithm is a pipeline of an A* graph search and a discrete variational method. The purpose of A* is to provide a good initial seed for the variational method, which guarantees that the final solution is locally optimal. The advantage of HADAD over pure graph search lies in the fact that it explores over a continuous space of trajectories. We describe next the two components of the pipeline in detail.

3.1. A* graph search

Graph optimization is a powerful mathematical technique widely used in the field of weather routing (Zis et al., 2020; Grifoll et al., 2022). These algorithms represent the ocean as a graph and aim to find the path that minimizes a specific objective function. The objective function can be travel time, fuel consumption, or other operational costs, depending on the requirements of the application. One advantage of graph search algorithms is their ability to easily incorporate constraints, such as obstacle avoidance, by disallowing certain nodes in the graph. Another advantage is their flexibility in adapting to any kind of cost function. However, this flexibility requires discretizing the search space, resulting in routes with abrupt turns, which are impractical for real-world applications. Moreover, searching in a discrete space will always provide sub-optimal solutions with respect to the relaxation problem where routes are allowed to be continuous trajectories on the surface of the sphere. Nevertheless, the trajectories obtained from graph search serve effectively as initial solutions for more refined optimization methods, as is the case for the FMS algorithm that we will introduce in Section 3.2.

3.1.1. Grid resolution

To implement graph optimization, we first need to discretize the world map using a grid. Each node of the grid corresponds to a coordinate in latitude and longitude. A practical choice for this grid is the one provided by our meteorological data, where the spacing between nodes in the ocean currents data is $1/12° \simeq 0.083°$ (approximately 10 km at the equator). This results in a grid of size 4320×2041, containing near 9 million nodes. It avoids points on land, as there are no currents or waves data there. This grid would be converted into an undirected graph, where each node represents a coordinate, and the edges connect adjacent nodes. However using this kind of square grid, has limitations due to the Earth's spherical shape, which requires significant distortion to fit the grid.

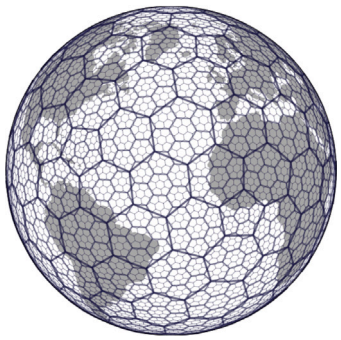


Fig. 2. Partition of the Earth into hexagonal grids of different sizes, using H3. Image from Uber Technologies, Inc. (2018).

Table 2
H3 resolutions.
Source: Data from Uber Technologies, Inc. (2018).

H3 Res.	Hexagon Area (km^2)	Hexagon Edge Length (km)	Number of unique indexes
3	12,392	59.8	41,162
4	1,770	22.6	288,122
5	253	9.8	2,016,830

To address this issue, we use instead an hexagonal grid provided by the H3 library developed by Uber Technologies (Uber Technologies, Inc., 2018). As shown in Fig. 2, hexagons mostly cover the round shape of the Earth, known as a hexasphere.

To generate this structure, the globe is first divided into icosahedrons, each containing 10 cells. However, since it is not possible to completely cover the globe with hexagons, 12 pentagons must be added to each icosahedron corner, resulting in a total of 122 cells at the lowest resolution. From this base, the resolution can be increased by adding 7 hexagons per cell. Although a perfect division is not achieved, this approach is sufficient to cover the space and create a hierarchical parent–child structure. Additionally, pentagons are only repeated at the center of parents that were originally pentagons, making it highly unlikely to encounter such polygons at higher resolutions.

There are different sizes of hexagons ranging from 4×10^6 km^2 to 1 m^2 with 15 resolution levels. Our A* graph search algorithm will be tested on grid resolutions ranging from 3 to 5. The specifics of these resolutions are shown in Table 2.

3.1.2. Edges of the graph

Each hexagon of the H3 grid has six neighbors. Compared to a traditional square grid, this hexagonal grid allows for two more possible directions at the same distance per node. This is important for real applications of weather routing, as vessels take smooth turns. A square grid forces 90° turns, while a hexagonal grid reduces that to 60°. However, that is still a sharp turn for maritime standards.

To address this challenge, we create new edges that connect each node to its ***K*-order neighbors**, following Mannarini et al. (2016). This increases the number of possible directions and reduces the angle between possible courses over ground, as depicted in Table 3. It is important to note that these figures are approximate and hold primarily in the planar approximation around hexagonal cells.

It should be noted that even in a planar hexagonal mesh, the minimum course correction is not isotropic. For instance, with $K = 3$, the minimum angle between certain directions can be as small as 11° rather than the nominal 20°. On a spherical surface, the situation becomes more complex due to the presence of pentagonal elements in the grid, which are necessary to ensure complete coverage of the

Table 3
Graph characteristics dependence on the K-order neighbors. The minimum course correction angle is approximate and not isotropic.

K	Maximum number of available neighbors	Minimum course correction
1	6	60°
2	18	30°
3	36	20°

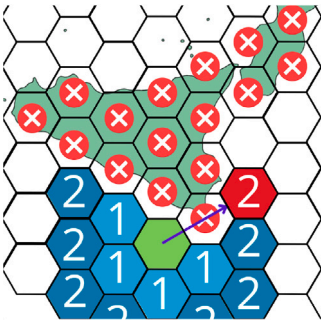


Fig. 3. The representation demonstrates how the algorithm can jump over several neighboring nodes, enhancing the flexibility of the route. Numbers on each node indicate the K-order of that neighbor to the origin (green node). It also shows the two techniques to ensure land avoidance: hexagons located directly on land are removed (marked with "x"), and edges that cross land are also removed (see the arrow connecting to the red node).

sphere. However, in practice, there are only 12 pentagons among over 40,000 hexagons at a grid resolution of 3, constituting less than 0.03% of the cells (Uber Technologies, Inc., 2018). Consequently, the impact of these pentagons on the overall routing algorithm is negligible, and it is reasonable to omit their consideration in our analysis.

One of the main challenges faced by weather routing algorithms is avoiding land. To ensure land avoidance, we implement two rules when building the graph: (1) hexagons located directly on land are removed; and (2) edges that cross land are also removed. This is shown in Fig. 3 for first and second order neighbors.

Finally, the weight of an edge connecting two nodes N and P is the time δt that the ship takes to move between them (in seconds):

$$g(N,P) = \delta t_{(N,P)} = \frac{\delta x_{(N,P)}}{\mathrm{SOG}_{(N,P)}} \tag{16}$$

We see that Eq. (16) is directly derived from Eq. (14). Thus, the weight of edge (N,P) depends on the Haversine distance between its nodes (δx) and the required speed over ground (SOG) to reach from N to P. The estimation of $\mathrm{SOG}_{(N,P)}$ requires a spatial interpolation, as the oceanographic conditions between nodes N and P vary. We interpolate the speed over ground in the following way:

$$\mathrm{SOG}_{(N,P)} = \frac{\mathrm{SOG}(\mathbf{x}_N, t_N) + \mathrm{SOG}(\mathbf{x}_P, t_N)}{2} \tag{17}$$

where $\mathrm{SOG}_{(N,P)}(\mathbf{x}, t)$ is the speed over ground required to go from N to P, under the oceanographic conditions at coordinates $\mathbf{x} = [u, v]$ and timestamp t. Note that t_P, the timestamp at which we reach node P, is unknown until we compute $g(N,P)$, the weight between nodes N and P. For this reason, we retrieve oceanographic conditions at node P at the same timestamp as node N, namely t_N. In order to compute the magnitude of the speed over ground, we work backwards from Eq. (13).

As the calculation of any weight $g(N,P)$ depends on the timestamp t_N, which in turn depends on previous nodes, the complete graph cannot be created in advance. Consequently, as nodes are closed, their times are fixed so they can be used when expanding new ones. This is also advantageous, as with A*, it is expected that not all nodes will need to be explored, which significantly reduces computation time.

3.1.3. Heuristic

With the graph constructed, the goal of our weather routing algorithm is to find the minimal path between the origin node O and the destination node D. The minimal path is defined as the one with the least travel time. The most popular method used in graph optimization problems is Dijkstra's algorithm (Dijkstra et al., 1959). This algorithm computes the minimal path between each pair of nodes, requiring exploration of all the edges within a graph. However, in this case, exploring the entire graph is not feasible due to its size (over 10 million nodes). An alternative is the A* algorithm (Hart et al., 1968), which unlike Dijkstra's algorithm does not need to explore all the graph's edges to find the minimal path between two nodes, thanks to the use of an heuristic inside the cost function. A* defines the cost function for node N as:

$$f(N) = g(O,N) + h(N), \tag{18}$$

where $g(O,N)$ is the cost of the path from O to N (a sum of edges connecting O to N), and $h(N)$ is the heuristic cost of the path from N to D. Following Huang et al. (2023), we choose

$$h(N) = \frac{\delta x_{(N,D)}}{\mathrm{SOG}_{(O,D)}}, \tag{19}$$

where δx is the Haversine (or great circle) distance and $\mathrm{SOG}_{(O,D)}$ is the average speed over ground computed along the route of shortest distance (orthodromic).

There are several variations of A* that can help the algorithm to converge faster. One of them is weighted A* (Ebendt and Drechsler, 2009), where the cost function is defined as:

$$f(N) = g(O,N) + w \cdot h(N). \tag{20}$$

Here, w is a weight that multiplies the heuristic component, affecting the number of nodes explored by the algorithm. For example, setting $w = 0$ cancels the heuristic component, making A* behave like Dijkstra's algorithm. Conversely, assigning a very high value to w causes the path from O to N to be disregarded, resulting in a Greedy Best First Search (Frasinaru and Raschip, 2019). Higher values of w are expected to speed up the algorithm, but no longer guarantee an optimal result, even when using admissible heuristics.

3.1.4. Consistency

To ensure the reliability of the A* search, two key aspects must be addressed. First, the grid resolution and K-order significantly impact the distance between waypoints $\delta x_{(N,P)}$. At the lowest resolution with a third-order neighbor configuration, $\delta x_{(N,P)}$ can reach up to 180 km, as derived from Tables 2 and 3. Such distances violate the assumption proposed in Section 4, which states that the distance between waypoints should be small enough (typically within 10 km) for oceanographic conditions to remain locally consistent (E.U. Copernicus Marine Service Information (CMEMS), 2024).

This issue was first addressed by Mannarini and Carelli (2019), who observed that in graph-based path-finding algorithms using regular grids, increasing the angular resolution by connecting to higher-order neighbors also increases the distance between connected nodes. This increase in distance can lead to decreased accuracy in estimating arc weights, as environmental conditions may vary significantly over longer distances. To overcome this, Mannarini and Carelli (2019) derived a constructive formula that allows for increasing the angular resolution without degrading the path resolution by refining the graph mesh spacing. By ensuring that the length of the arc leading to a specific angular resolution remains constant, they maintained consistent accuracy in the estimation of arc weights.

In our study, we face a similar challenge. The distance between waypoints $\delta x_{(N,P)}$ can become excessively large when using higher-order neighbor configurations on a coarse grid. To address this, we ensure that $\delta x_{(N,P)} \leq 10$ km by introducing intermediate points along the edge connecting nodes N and P. Specifically, if $\delta x_{(N,P)}$ exceeds

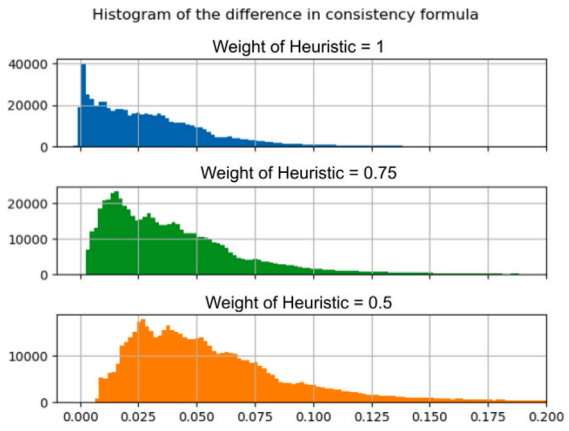


Fig. 4. Histogram of the distribution of Δh values for different weights of the A* heuristic, across 20 different routes, resulting in 614,494 evaluations. A consistent heuristic is one that ensures $\Delta h(N) \geq 0, \forall N$. As the weight of the heuristic decreases, the distribution shifts right, indicating an increase in the consistency of the heuristic.

10 km, we recursively subdivide the segment by placing new points at midpoints until all segments between consecutive points are less than 10 km. This approach effectively refines the path resolution without altering the overall grid resolution, allowing us to maintain accuracy in arc cost estimations while benefiting from increased angular resolution provided by higher-order neighbor connections.

The total cost of the edge (N, P) is then calculated as the sum of the costs between these intermediate points. By breaking the edge into smaller segments, we ensure that the cost estimates in the A* search remain consistent and reliable, regardless of grid resolution or K-order neighbors. This approach also preserves the land avoidance feature of A* by maintaining the same path constraints throughout the process.

With reliable cost computation established, the next concern is the heuristic. A* will always find an optimal solution as long as the heuristic is consistent. For a heuristic to be consistent, it must satisfy expression (21) below

$$h(N) \leq g(N, P) + h(P), \tag{21}$$

where N is a node in the graph, P is a node expanded from N, h is the heuristic function, and $g(N, P)$ is the actual cost of traveling from N to P. In our case, theoretically validating the consistency of the heuristic is challenging due to the variability of meteorological data and its impact on the vessel. Therefore, an empirical study was conducted to explore the consistency of the heuristic.

The consistency condition was tested for each node explored across 20 different routes, resulting in 614,494 evaluations. Fig. 4 shows the relative variation between $h(N)$ and $g(N, P) + h(P)$, expressed as:

$$\Delta h(N) = 1 - \frac{h(N)}{g(N, P) + h(P)} \tag{22}$$

A perfect heuristic would require that $\Delta h(N) = 0, \forall N$. However, for the heuristic to be consistent, it is sufficient that the relative differences remain non-negative: $\Delta h(N) \geq 0, \forall N$. With $w = 1$, the heuristic is inconsistent in only 1.01% of cases, meaning that $\Delta h(N) < 0$ in 1 out of every 100 nodes opened. However, reducing the weight to $w = 0.75$ or $w = 0.5$ ensures consistency. Since reducing the weight slows down the search, it is crucial to find a balance between lowering the weight to maintain heuristic consistency and maintaining search speed.

Ensuring that the heuristic is consistent is crucial for the time-dependent A* algorithm because it prevents the reopening of already closed nodes while still obtaining an optimal solution. Zhao et al. (2008) state that A* can be applied similarly to the original algorithm as long as the FIFO condition is met:

$$t_1 + h(N, t_1) \leq t_2 + h(N, t_2), \tag{23}$$

where t_1 and t_2 are two time instances such that $t_1 \leq t_2$, $h(N, t_1)$ is the heuristic of node N at time t_1, and $h(N, t_2)$ is the heuristic of N at time t_2. In our case, since the heuristic is not time-dependent (19), this expression simplifies to $t_1 \leq t_2$, which is already guaranteed by the initial condition. This simplification of the algorithm significantly reduces the computation time.

3.2. FMS variational algorithm

Due to the discrete space in which graph search optimizes, the A* algorithm generates a path based on the hexagonal grid structure. As outlined in Table 3, even when using third-order neighbors, the algorithm's course correction is limited to a minimum angle of 20°. Such abrupt changes are suboptimal for shipping vessels, which benefit from smoother course adjustments to ensure efficient and safe navigation. To address this limitation, after the A* search we apply the Ferraro–Martín de Diego–Sato (FMS) algorithm (Ferraro et al., 2021, 2022) to refine the solution. FMS solves numerically a Boundary Value Problem (BVP) associated with Zermelo's Navigation Problem, which smoothens course corrections and further improves overall route efficiency.

The objective of the FMS algorithm is to minimize the total travel time by performing local adjustments to the path, derived from the following cost function:

$$T(\mathbf{x}_i, \mathbf{x}_{i+1}) = \frac{2 \cdot \delta x_i}{\mathrm{SOG}_i + \mathrm{SOG}_{i+1}} \tag{24}$$

In this equation, δx_i represents the Haversine distance between two consecutive waypoints $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$, while SOG_i and SOG_{i+1} denote the speeds over ground at these points. Both speed modules are estimated under the oceanographic conditions at their respective coordinates and timestamps t_i and t_{i+1}. We recompute the timestamps after each update to ensure kinematic consistency, as explained below. Eq. (24) is functionally the same as the weight of A* edges defined by Eq. (16), replacing the nodes in a discrete graph by points in a continuous space.

As initial seed, FMS takes a route defined by M points $\{\mathbf{x}_i\}_{i=1}^{M}$ where $\mathbf{x}_1$ and $\mathbf{x}_M$ are the fixed coordinates of the origin and destination ports, and the rest of the points $\{\mathbf{x}_i\}_{i=2}^{M-1}$ can be modified. Remember that $\mathbf{x}_i = [u_i, v_i]$, where u is the longitude coordinate and v is the latitude. In HADAD, the initial seed for FMS is the output trajectory of A*. This route ensures land avoidance and its waypoints are at most 10 km apart, to ensure that the constant weather approximation in each segment is sound.

FMS iteratively moves the points of the initial seed $\{\mathbf{x}_i\}_{i=2}^{M-1}$ to obtain a new trajectory $\{\mathbf{x}_i^*\}$ that minimizes the overall cost $\sum_{i=1}^{M-1} T(\mathbf{x}_i^*, \mathbf{x}_{i+1}^*)$. At each iteration, the algorithm performs local optimization by adjusting the spatial positions of the waypoints to reduce the total travel time, while ensuring kinematic consistency by updating timestamps accordingly. Specifically, the algorithm solves the following equation for $\{\mathbf{x}_i'\}$:

$$\begin{aligned} &D_2 T(\mathbf{x}_{i-1}, \mathbf{x}_i) + D_1 T(\mathbf{x}_i, \mathbf{x}_{i+1}) \\ &+ \left(D_{22} T(\mathbf{x}_{i-1}, \mathbf{x}_i) + D_{11} T(\mathbf{x}_i, \mathbf{x}_{i+1})\right)(\mathbf{x}_i' - \mathbf{x}_i) = 0, \end{aligned} \tag{25}$$

followed by an update $\{\mathbf{x}_i\} \leftarrow \{\mathbf{x}_i'\}$.

Here D_k represents the partial derivatives with respect to the kth argument of the cost function T, and $D_{k,l} = D_k^{\circ} D_l$. These partial derivatives are calculated numerically using reverse differences. For example (the other terms work similarly):

$$D_1 T(\mathbf{x}_{i-1}, \mathbf{x}_i) = \frac{1}{\epsilon} \begin{bmatrix} T\left(\mathbf{x}_{i-1} - \begin{bmatrix} \epsilon \\ 0 \end{bmatrix}, \mathbf{x}_i\right) - T(\mathbf{x}_{i-1}, \mathbf{x}_i) \\ T\left(\mathbf{x}_{i-1} - \begin{bmatrix} 0 \\ \epsilon \end{bmatrix}, \mathbf{x}_i\right) - T(\mathbf{x}_{i-1}, \mathbf{x}_i) \end{bmatrix}. \tag{26}$$

We use a difference step $\epsilon = 10^{-4}$ degrees in our experiments, which is sufficiently small compared to the typical distance between points, yet large enough to avoid numerical instability when using double floating-point precision. For a comprehensive explanation of the method's mathematical principles and detailed derivations, please refer to Ferraro et al. (2021).

FMS asserts that the above iteration eventually converges to the desired local optimum $\{\mathbf{x}_i^*\}$. However, since we rely on numerical derivation to approximate the D operators, in practice it is best to introduce a damping factor below 1 to ensure convergence:

$$\mathbf{x}_i' \leftarrow \mathbf{x}_i + \text{damping} \cdot (\mathbf{x}_i' - \mathbf{x}_i) \tag{27}$$

As an additional enhancement, to preserve the land avoidance of A*, we check the new trajectory $\{\mathbf{x}_i'\}$ for land intersections after each iteration. If any proposed waypoints cross land, they are reverted to their previous positions, $\mathbf{x}_i$.

After updating the coordinates, we recompute the timestamps of the new waypoints $\mathbf{q}_i' = [u_i', v_i', t_i']$. The travel time is recalculated iteratively from point $i = 1$ to $i = M$ using Eq. (24), ensuring that the vessel's speed and position are consistent at each point along the route. Once the new set of waypoints $\{\mathbf{q}_i'\}_{i=1}^M$ is determined, the next FMS iteration begins.

This process ensures that the vessel's kinematic integrity is maintained throughout the optimization. By updating both spatial positions and timestamps, we avoid any inconsistencies that could arise from adjusting positions without accounting for time.

It is important to note that the FMS algorithm refines the initial path provided by the A* algorithm, improving smoothness and efficiency, without challenging the global optimization achieved by A*. The A* algorithm provides a globally optimal path within the discretized search space, and FMS enhances this path through local adjustments in a continuous space.

3.2.1. FMS parameters

The FMS algorithm comprises three parameters:

- **Damping**, motivated in (27), is akin to the *learning rate* in gradient descent. It dictates the step size taken towards a local minimum at each iteration, and thus balances rate of convergence and risk of overshooting (Buduma and Locascio, 2017).
- An **early stopping** mechanism which halts the algorithm if the cost function does not decrease for a specified number of consecutive iterations. This aims to avoid unnecessary computations once the local minimum is near. After stopping the algorithm, it will output the solution with the lowest cost that has been reached.
- We also set a **maximum number of iterations** to ensure that in any case the algorithm runs in reasonable time.

Further details can be found in Precioso et al. (2024) and the references therein.

4. Case studies

Having introduced our optimization algorithm, we can now define the set of instances that make the benchmark where we test HADAD. We will need to define the ODPs, departure dates, vessel parameters and operational characteristics. The cost function has already been introduced in Section 2.

Selecting a good set of ODPs for weather routing requires attention, as it first need to be representative of real world applications. In Table 1, all ODPs reflected real shipping routes. It also need to be of interest to test various weather routing algorithms to their full capacity. This not only requires weather routing algorithms to find an optimal path according to some criterion, but also consider obstacle avoidance. These ODPs should therefore appear in different levels of complexity, including land presence and difficult meteorological conditions.

To meet the first criteria, we can take a look at maritime networks. A maritime network is a system of maritime flows and connections between ports and other locations (Liu et al., 2023). Maritime networks reveal various aspects of global shipping, including mapping traffic densities to show the spatial patterns of vessel movements (Ducruet, 2020). Via the density of maritime networks, we can identify the major hubs: the ports that are most visited. We have chosen a set of instances based on these major hubs, as they will represent ODPs relevant for the shipping industry. A literature review has been conducted to extract the major hubs in recent maritime shipping (Alderson et al., 2020; Álvarez et al., 2021; Ge et al., 2022).

Next step is to ensure that our instances include ODP with several levels of complexity. The ODPs for the five chosen shipping routes are shown in Table 4. Each of these five routes will be considered in forward and reverse direction. As for the departure dates, we consider a departure for every Sunday of 2023, for a total of 52 departure dates per ODP. This ensures a variety of scenarios, as the ocean conditions change with the seasons.

For the vessel type, this benchmark will use a typical container ship. Its parameters are listed in Table 5. We assume the vessel can achieve constant power delivery for the entire duration of the journey, and as such, the vessel sails at constant speed in calm water, SCW. We considered three different SCW, at very slow (6 knots), slow (12 knots), and normal speed (24 knots).

Together with the 52 departure dates, ten ODPs, and three different speeds with respect to the water, we have a total of 1,560 instances, providing a comprehensive basis for the purposes of this study. The goal for all instances will always be to minimize the travel time, accounting for the effect of waves and ocean currents as explained in Section 2. Solutions for every route will be reported as polygonal curve with waypoints $\{\mathbf{q}_i\}_{i=1}^M$ where $\mathbf{q}_i = [u_i, v_i, t_i]$. The optimization algorithm must ensure that the average distance between two consecutive waypoints is less than 10 km, which is of the same order as the grid spacing of weather data (E.U. Copernicus Marine Service Information (CMEMS), 2024). This ensures that the hypothesis that weather conditions remain constant over each segment of the route is sound.

For an easier interpretation of the results, all solutions will be compared with a reference standard route which is given with the instance. This standard route for comparison is chosen to be the shortest distance route (also referred to as *orthodromic*). Thus, rather than giving the total time of a given candidate solution, we will report the relative reduction (or increase) in travel time of the candidate route with respect to the shortest distance route departing on the same moment.

4.1. Weather data

The analysis of the vessel motion to calculate the travel time needs to have access to weather data, specifically ocean currents and waves. For this study, we used real weather data, particularly the NEMO model for currents data and the MFWAM model for waves. A full explanation of these two models and the processing applied to them is provided in Appendix B.

The most typical interpolation methods in a 2-D setting are Nearest Neighbor, Bilinear, and Bicubic. Nearest Neighbor and linear methods are the simplest and the fastest to compute. In fact, they would be perfectly valid for most optimization algorithms, such as a pure A* that only requires access to the values of the speed function. However, these methods are not suitable for the variational optimization methods, such as the FMS we will present in Section 3.2. Indeed, that family of methods compute derivatives of the velocity, which requires a smooth dependence on the weather external fields. Nearest Neighbor and linear interpolation methods are not everywhere differentiable, thus rendering variational algorithms computationally unstable. For the sake of fairness across weather routing algorithms, we introduce bicubic interpolation, which is computationally more expensive but produces a smooth interpolated field. We further discuss the implementation of this interpolation method in Appendix D.

Table 4
Origin–Destination Pairs: regions they connect and orthodromic distances. Each ODP yields two possible routes: from port 1 to port 2, and vice versa. The column "N. Nodes" is relevant to the A* algorithm and will be used in Section 5.5.

Port 1 (Code)	Port 2 (Code)	Connection	Ocean	Distance	N. Nodes
Hamburg (DEHAM)	New York (USNYC)	Europe - America	Atlantic Ocean	6248 km	8705
New York (USNYC)	Colón (PAONX)	Atlantic - Panama	Atlantic Ocean	3617 km	3048
Balboa (PABLB)	Callao (PECLL)	Panama - Pacific	Pacific Ocean	2474 km	1273
Kuala Lumpur (MYKUL)	Hurghada (EGHRG)	Asia - Suez	Indian Ocean	8416 km	8689
Said (EGPSD)	Algeciras (ESALG)	Suez - Europe	Mediterranean Sea	3533 km	1240

Table 5
Parameters of every instance in our benchmark.

Journey	
ODP	See Table 4 (10 ODPs).
Departure Date	Every Sunday of 2023, first one being 00:00 01-01-2023 UTC and last one being 00:00 24-12-2023 UTC (52 departure dates).
Vessel	
Length	$L = 220$ m
Displacement	$\nabla = 36500$ m^3
Block Coefficient	$c_B = 0.6$
Speed Calm Water	$SCW = \{6, 12, 24\}$ kn
Optimization Problem	
Cost Function	Travel time affected by waves and currents.
Solution	Polygonal curve defined by a set of waypoints $\{\mathbf{q}_i\}_{i=1}^{M}$

5. Results

After performing a parameter search for the first week of each instance, we selected the configuration of our algorithm by considering the trade-offs between compute time and relative route savings. We relegate the discussion of the parameter tuning of HADAD to Appendix E. The A* algorithm used for the remainder of this paper is configured with the following parameters: grid resolution of 4, third-order neighbors, and a heuristic weight of 0.5. The output is then smoothed by the FMS algorithm with no damping, for a maximum of 2,000 iterations, and is subject to early stopping if no improvements are observed in the previous 20 consecutive iterations. All experiments were conducted on the machine specified in Appendix E.

We can visualize an example of the solutions produced by HADAD in Fig. 5. This figure shows a map with the orthodromic route (purple), the A* path (yellow), and the HADAD optimized route (red) for the journey from New York to Panama (USNYC-PAONX), departing on January 1st, 2023, at a speed of 6 knots. The yellow line represents the path generated by the A* graph search, which exhibits sharp changes in direction along the route due to the grid structure. After applying several hundred iterations of the FMS algorithm, the red line shows that these abrupt turns have been smoothed, resulting in a more navigable and efficient route.

There are five pairs of ports, introduced in Table 4, that can be traveled in both directions, totaling ten ODPs. Departures occur every Sunday of 2023, starting on January 1st, resulting in 52 departures per ODP and vessel speed. The vessel model is the same for all instances. Sailing speed with respect to the water is constant along a journey, and three different speeds will be studied: 6 knots, 12 knots, and 24 knots. This setup results in 1,560 experiments. To facilitate further analysis and discussion, we group the experiments to study different effects.

5.1. Vessel speed

The first study conducted with these results examines how the gains of our HADAD algorithm are affected by vessel speed. Fig. 6 shows a histogram of gains across all instances based on vessel speed through water. Each histogram represents the distribution of time

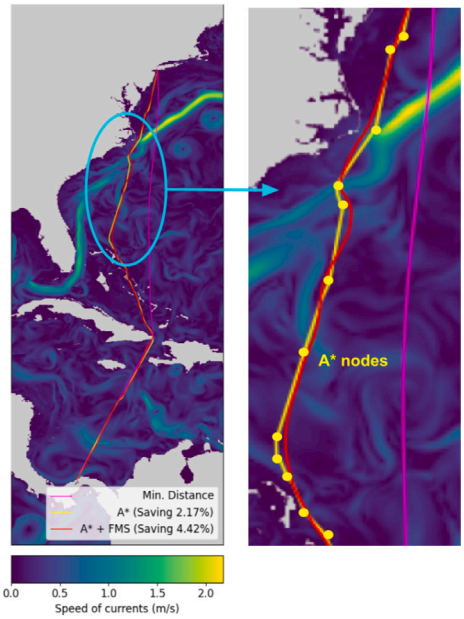


Fig. 5. Routes from New York to Panama, departing on January 1, 2023, at a sailing speed of 6 knots. The left panel compares the orthodromic route (purple), the route generated by the A* algorithm (yellow), and the optimized route produced by HADAD (red). The HADAD route demonstrates how applying the FMS algorithm smooths out the 'jagged' turns of the A* route, transitioning from the discrete grid-based path to a continuous trajectory. The right panel zooms in to highlight the effect of the A* grid on the jagged turns, showing how the route connects the graph nodes (marked with yellow dots).

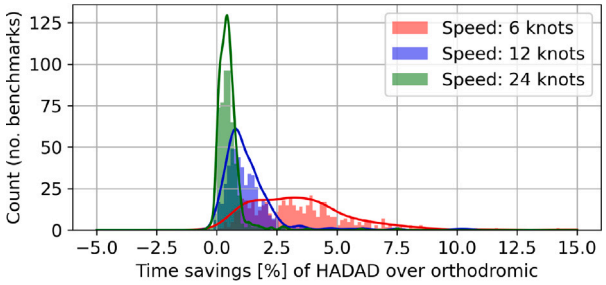


Fig. 6. Time savings of our HADAD algorithm compared to the orthodromic route. Histogram and best-fit KDE univariate distribution for each sail speed.

savings achieved by HADAD compared to the orthodromic route at a specific sailing speed (6, 12, and 24 knots). The results indicate that lower vessel speeds achieve greater gains with weather routing. At a vessel speed of 6 knots, time savings average 3.60% (with a standard deviation of 2.61). For 12 knots, savings are 1.36% (1.61), and for 24 knots, they decrease to 0.51% (0.58).

Note that, at higher speeds, there are instances in which the HADAD solution does not reduce travel time with respect to the route of shortest distance. Specifically, when sailing at 12 knots, seven out of 520 experiments show negative gains. At 24 knots, 14 experiments yield worse

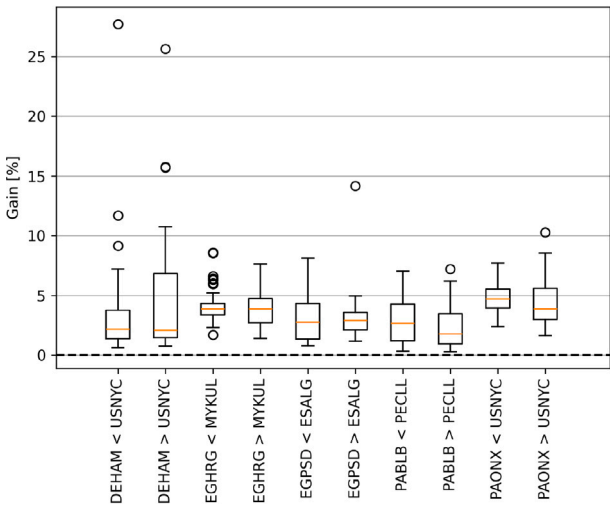


Fig. 7. Percent reduction in travel time achieved by the HADAD algorithm compared to the orthodromic route at 6 knots. ODPs are sorted by decreasing distance along the X-axis, and formatted such that the arrow ">" always points to the destination, i.e., "D < O", "O > D".

results with HADAD compared to following the orthodromic route. These losses are marginal, the lowest being −0.4% (20 min longer than the shortest distance for an 84-hour journey). However, these cases are still worth investigating as they represent very challenging scenarios and/or shortcomings of our HADAD algorithm. In contrast, at lower speeds, HADAD can save up to 27% of travel time in some scenarios. We will explore these extreme cases in detail later.

Overall, these results highlight the significant impact of vessel speed on weather routing algorithms. In our dynamic models, the impact of weather grows as cruising speeds decrease.

5.2. Geographical variation

Our next discussion focuses on how the gains from weather routing depend on the choice of Origin–Destination Pair (ODP). As noted in Table 1, each paper employs a different set of ODPs, making it essential to assess whether this choice affects the results reported by the algorithm. In Fig. 7, we compare the overall gains (travel time reduction relative to the orthodromic route) of our HADAD algorithm for each set of ODPs, sorted by direction of travel, when sailing at 6 knots. Fig. 7 presents boxplots of the percentage reduction in travel time achieved by HADAD compared to the orthodromic route for each ODP. The ODPs are sorted by decreasing distance along the X-axis, and each boxplot summarizes the distribution of savings over 52 weeks, showing the median, quartiles, and outliers. The general trend of these boxplots is similar for vessel speeds of 12 and 24 knots, with gains inversely proportional to speed.

We first observe that median gains (orange line, Q50) differ between pairs of ports. From lowest to highest gains:

- 1.9% for Balboa (PABLB) - Callao (PECLL),
- 2.1% for Hamburg (DEHAM) - New York (USNYC),
- 2.5% for Port Said (EGPSD) - Algeciras (ESALG),
- 3.7% for Hurghada (EGHRG) - Kuala Lumpur,
- 4.2% for Colón (PAONX) - New York (USNYC).

The distribution of gains during the whole year has significant outliers, especially in the route across the North Atlantic (DEHAM-USNYC). These are due to extreme weather events and will be analyzed in Section 5.3.

The presence of strong oceanographic conditions, such as the Gulf Stream in PAONX-USNYC, increases the potential gains of weather routing. We also observe that gains tend to increase with the distance between ODPs, as this allows the algorithm to explore a broader space (refer to Table 4 for distances). Indeed, the longest routes present the largest outliers, demonstrating that the algorithm can exploit certain favorable conditions when the weather is advantageous.

Table 6
Gains of HADAD algorithm over orthodromic route, grouped by each season of 2023 in the Northern Hemisphere.

Season	Avg. Gain % (Std.)		
	6 kn	12 kn	24 kn
Winter	5.02 (3.69)	2.09 (2.61)	0.71 (0.93)
Spring	4.06 (1.75)	1.42 (0.75)	0.53 (0.31)
Summer	2.74 (1.84)	0.96 (0.94)	0.39 (0.36)
Autumn	2.53 (1.87)	0.99 (1.28)	0.39 (0.49)

In Section 5.1, we noted that HADAD would sometimes not outperform the orthodromic route at higher speeds. We now see that this effect is dependent on the ODP: at 24 knots, PABLB-PECLL shows eight instances out of 52 with negative gains, PAONX-USNYC shows four, while EGPSD-ESALG and DEHAM-USNYC have only one each, and none in EGHRG-MYKUL.

These findings emphasize the importance of carefully selecting ODPs when evaluating the performance of weather routing algorithms. Variations in oceanographic conditions, route length, and external factors significantly influence the potential gains, highlighting the need for a diverse set of instances to accurately assess algorithm performance.

5.3. Seasonal study

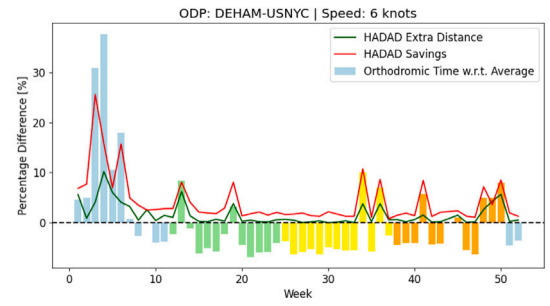
We have assessed how vessel speed and the ODP affect the overall gains achievable with weather routing. Next, we discuss the seasonal effect on the gains of a journey. We group our instances by seasons in the Northern Hemisphere, as shown in Table 6.

We identify greater savings across all speeds during winter, and less so in spring, compared to summer and autumn. In fact, gains in winter almost double those in other seasons. The standard deviation of winter is also the highest, implying that this is the season with the largest weekly differences, i.e., the most variability.

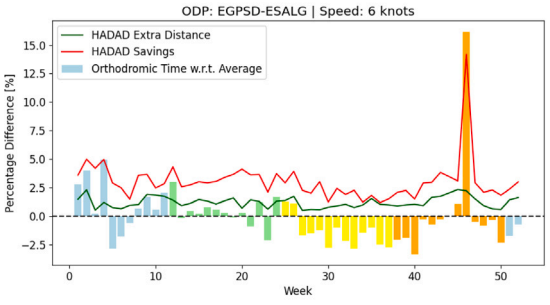
To better assess the impact of seasonality, we focus on how navigation conditions affect the fixed shortest distance route across the year. Fig. 8 illustrates the percentage difference in travel time between the orthodromic route and the yearly average for each week of 2023, along with the gains achieved by HADAD and the extra distance covered. The data are presented for three ODPs: Hamburg to New York (Fig. 8(a)), Suez Canal to Algeciras (Fig. 8(b)), and New York to Panama Canal (Fig. 8(c)). In each subfigure, the vertical bars show the relative variation of travel time for the orthodromic route with respect to the yearly average, colored by season. The solid lines represent the gains of HADAD (red line) and the extra distance covered (green line) for every week.

Fig. 8(a) represents Hamburg (DEHAM) to New York (USNYC), crossing the North Atlantic Ocean. During winter (colored in blue), we see a significant increase in travel time for the orthodromic route due to harsher weather conditions. In particular, during the third and fourth week, we observe an increase in travel time of around 35% above the yearly average. With such a great discrepancy in time, HADAD is able to achieve increased savings of nearly 25%. This is only evident in the first few weeks of the year. In spring, summer, and fall, most orthodromic routes perform consistently, with similar savings from HADAD. These variations are largely due to the fact that the routes between DEHAM and USNYC are in the North Atlantic, where studies have observed extreme wave climates in winter (Morales-Márquez et al., 2020; Gleeson et al., 2019).

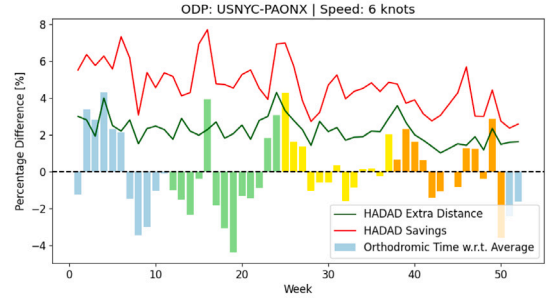
Such extreme wave climates are also observed in the Mediterranean Sea. This fact is reflected in Fig. 8(b) for Egypt (EGPSD) to Spain



(a) Hamburg - New York



(b) Suez canal - Algeciras



(c) New York - Panama canal

Fig. 8. Percentage difference between the orthodromic route per week to yearly average, and HADAD compared to orthodromic with respect to time and distance. Each week is colored by season, starting with winter (1st January, 2023).

(ESALG), where an extreme event during summertime causes one orthodromic route to take 15% more time than usual, allowing HADAD to achieve savings on the same scale. In contrast, routes in the Caribbean Sea show small fluctuations in travel time across the year, without a clear seasonal tendency. This is shown in Fig. 8(c) for the route between New York (USNYC) and Panama (PAONX).

In summary, this seasonal study shows that travel time is on average roughly constant across the year, with a variance around 3%. There exist, however, extreme weather events that can drastically increase the sailing time for a given route. Weather routing algorithms such as HADAD are able to anticipate these adverse conditions by planning beforehand, with remarkable improvement on travel time. Even in conventional circumstances (when no extreme event is present), our HADAD algorithm consistently finds routes that are on average 3% faster than the standard geodesic route.

5.4. Importance of waves and currents

To conclude this discussion, we find it relevant to study the separate effect of waves and currents in the weather routing optimization problem. This is, of course, strongly dependent on the vessel characteristics and the resistance model, but some interesting lessons can be learned.

Table 7
Comparison between considering different weather variables, percentage gains (standard deviation) with only currents, only waves, and both effects.

Speed (knots)	Gains avg. (std.) (%)		
	Only Waves	Only Currents	Both
6	0.47 (2.03)	3.10 (1.72)	3.55 (2.58)
12	0.35 (1.53)	1.06 (0.59)	1.34 (1.59)
24	0.12 (0.55)	0.42 (0.25)	0.50 (0.58)

We know from the equations in Section 2 that waves can reduce vessel speed by up to 70% in adverse conditions. Likewise, strong ocean currents can reach speeds up to five knots (National Oceanic and Atmospheric Administration (NOAA), 2024c). Following these prior assumptions, we expect waves to have a larger impact, while currents become more relevant for slow steaming.

We consider thus three separate speed reduction models that consider only waves, only currents, or both. We compare the gains of our HADAD algorithm for all of these experiments in Table 7.

We found that for every vessel speed, the model that takes into account only waves achieves a smaller reduction than the one that considers only currents, while it is natural that both are dominated by the model that considers both effects. However, the standard deviation of the HADAD gains for the wave-scenario is wider than the currents-scenario, implying the existence of extreme weather events where waves become very relevant. This fact is probably explained by the larger variation of wave conditions over time compared to ocean currents.

To study this effect in further detail, we consider orthodromic routes and represent the distribution of both Beaufort numbers and speed of ocean currents encountered along the routes. Fig. 9 shows histograms of the Beaufort number (Fig. 9(a)) and the ocean current speed relative to the ship's course over ground (Fig. 9(b)), based on all waypoints across every orthodromic route for all ten ODPs throughout the 52 weeks of the year 2023.

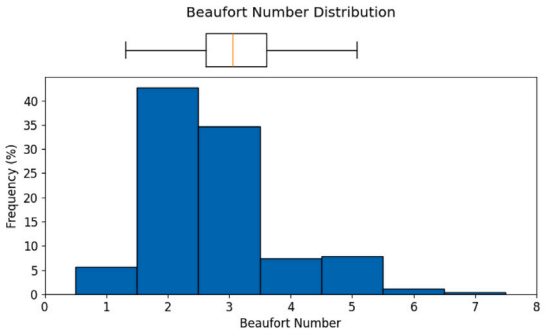
In Fig. 9(a), we see that 97% of the orthodromic routes' waypoints encounter a Beaufort number lower than five. The Beaufort number is strongly related to wave height by Eq. (5), and we can see its effect on speed reduction in Fig. 1, according to the model used in this study. The average speed reduction due to waves is minimal, around 0.3%—a percentage similar to the gains achieved by our HADAD algorithm when only waves are considered. In fact, even when BN = 5 (3-meter-high waves), the speed reduction is less than 5%. At a nominal sailing speed of 24 knots, this gives a net speed reduction of around 1.8 knots.

The speed of ocean currents is shown in Fig. 9(b). The effect of currents results in an average increase of 0.6% in the vessel's speed, reaching 5% at its maximum. This matches our results in Table 7, as gains over the orthodromic route are higher on average and more consistent (smaller standard deviation) with currents than with waves. A critical difference between both effects is that waves always subtract speed, while ocean currents can have both a positive or negative effect.

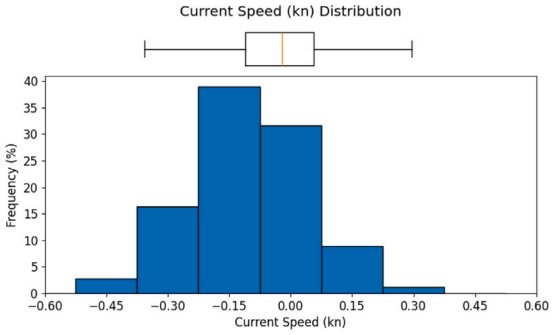
In summary, if the route encounters extreme weather events with Beaufort numbers higher than 6 (waves over 4 meters high), the effect of waves is very significant, and weather routing algorithms learn to avoid passage through these areas. In standard weather conditions, ocean currents become the most relevant factor, as they do not suffer strong variation in time.

5.5. Computational cost

It is also necessary to analyze the computational cost of HADAD, using the large number of experiments conducted. The ODPs shown in Table 4 have been divided into different categories based on the number of nodes: small (< 2,000 nodes), medium (< 5,000 nodes),



(a) Beaufort Number, BN.



(b) Ocean current speed relative to the ship's course over ground. Positive values represent favorable currents.

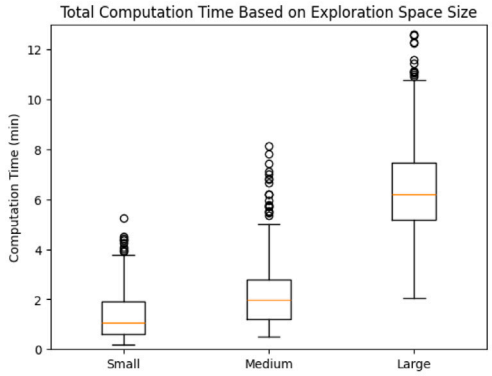
Fig. 9. Distribution of values encountered across every orthodromic route for all ten ODPs across the 52 weeks of the year 2023.

and large (< 10,000 nodes). In total, 1,560 experiments have been conducted for this study on computational cost.

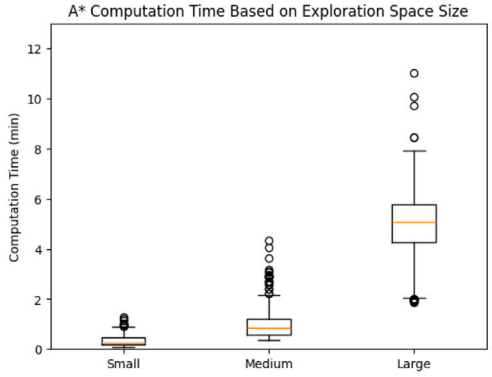
In Fig. 10, we observe that as the search space of HADAD increases, so does the computation time to solve the problem. The figure shows box plots of computation times for the A* algorithm, the FMS algorithm, and the total computation time, across different sizes of the exploration space. The graphs indicate that the total computation time follows the $O(|V|\log|V|)$ complexity of the A* algorithm, where $|V|$ is the size of the vertices set of the graph. When considering only the A* time, the same behavior is observed. On the other hand, for FMS the trend is more linear, indicating that the number of cells has little effect on the computation time of the variational algorithm. This is likely due to the algorithm updating all points in parallel, which explains why the difference between varying sizes is barely noticeable. Although FMS adheres to a complexity of $O(|V|)$, it has a slight slope. For more details on the time complexity of the method, please refer to Appendix C.

5.6. A specific example

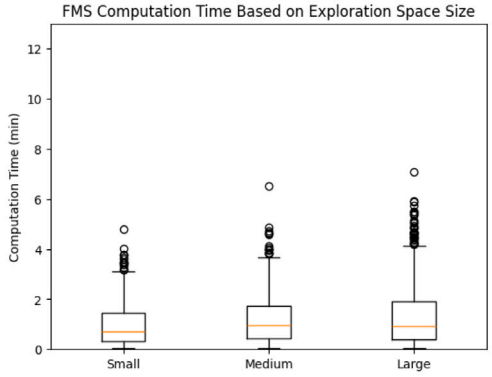
To conclude, we take a closer look at one of the instances with higher gains: the route from Suez Canal (EGPSD) to Algeciras (ESALG) starting on week 46 of 2023 and sailing at 6 knots. The shortest distance route for this instance takes 15% longer travel time than average, as shown in Fig. 8(b). Fig. 11 shows the percentage loss in sailing speed due to waves and ocean currents for both the orthodromic route and the one optimized using HADAD. In the figure, the vertical axis represents the percentage loss in speed, and the horizontal axis represents the progression along the route. Both routes face adverse weather conditions at the start of the journey. However, the HADAD solution takes a detour to follow favorable currents, reducing the speed loss caused by waves. In contrast, the standard route goes against strong currents (nearly 3 knots, half its sailing speed), leading to a loss of up



(a) Total computation time across all algorithms for varying sizes of the exploration space. As the exploration space increases, the total computation time generally increases, reflecting the expected behavior of $O(|V|\log(|V|))$ complexity.



(b) Computation time of A* for varying sizes of the exploration space. As the exploration space increases, the computation time increases, reflecting the expected behavior of $O(|V|\log(|V|))$ complexity.



(c) Computation time of FMS for varying sizes of the exploration space. This reflects a linear behavior following $O(|V|)$ complexity.

Fig. 10. Box plots showing the computation times (in minutes) for three categories of exploration space size (small, medium, and large).

to 50% of its original speed. This prolonged exposure to bad conditions forces the vessel to spend a longer time in an unfavorable area, adding to the delay. As a result, the shortest distance route is covered in 390 hours for a total distance of 3533 km (SOG ≈ 5 kn), while the HADAD optimized route covers 3611 km in 335 hours (SOG ≈ 6 kn).

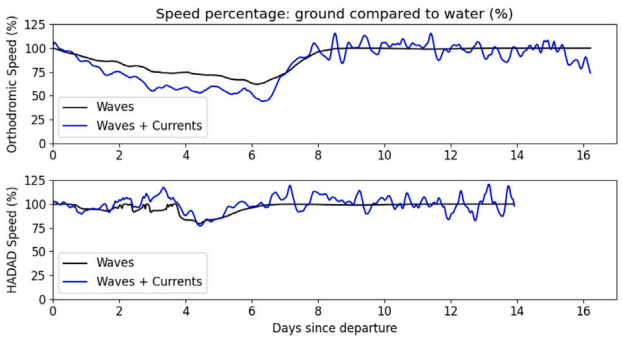


Fig. 11. Percentage loss in sailing speed due to waves and ocean currents for the route from Suez Canal (EGPSD) to Spain (ESALG) during week 46 of 2023, at a sailing speed of 6 knots. The figure compares the shortest distance route (above) and the optimized route using HADAD (below).

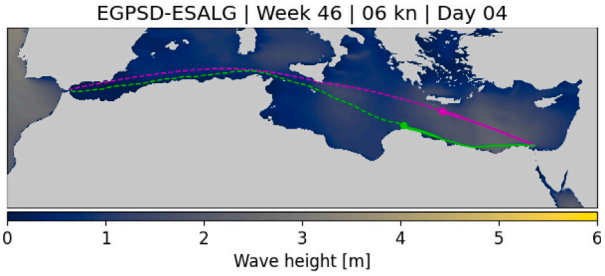


Fig. 12. Route from Suez Canal (EGPSD) to Spain (ESALG) during week 46 of 2023, at a sailing speed of 6 knots. The figure compares the orthodromic route (magenta) and the optimized route using HADAD (green).

Fig. 12 shows the trajectories traversed by both routes on a map. The orthodromic route (magenta) follows the shortest distance between the two ports, while the HADAD optimized route (green) takes a detour to avoid unfavorable conditions. The combined effect of waves and currents slows down the shortest distance route significantly, showing the benefit of weather routing to plan ahead.

Looking back at Table 7, a final observation to be made is that both waves and currents are relevant effects. When accounting for both, the gains achieved by HADAD are almost additive, and more variable (they have the highest standard deviation). This finding underscores the importance of taking every factor into account when developing a weather routing algorithm, and future versions will include more oceanographic and weather variables, as well as more complex and realistic models for the cost function.

6. Validation

In the previous sections, we have demonstrated the capabilities of HADAD in real-world scenarios. However, the lack of standardized benchmarking practices makes it challenging to directly compare our method with others. To address this, we validate HADAD using synthetic vector fields commonly employed in the literature. Testing algorithms on synthetic vector fields is crucial for several reasons. First, they provide a controlled and consistent environment, allowing for accurate and reliable assessments of algorithm performance under varying conditions. Second, synthetic vector fields can be precisely manipulated to introduce specific challenges, enabling us to test the robustness and reliability of the algorithm. This approach provides valuable insights into the limitations and potential improvements of the algorithm.

In this validation, we use several synthetic vector fields that have been proposed in previous research to test navigation algorithms. These include the Four-Vortices field and Swirlys field from Ferraro et al. (2021), and the time-dependent vector field from Techy (2011). Table 8 summarizes the vector fields, the optimization goals, and the results obtained.

To apply HADAD to these synthetic vector fields, we needed to make adjustments related to the computation of distances. Since we are transitioning from spherical coordinates to a Euclidean plane, the distance formula reflects this change. To ensure consistency in cost computation, as established in Section 3.1.4, we set $\delta x_{N,P} \leq 10$ km in the spherical case. Now, with Euclidean distances, we update the condition to $\delta x_{N,P} \leq 0.05$ units. All the other A* and FMS parameters remain the same as those used in Section 5.

6.1. Four-vortices vector field

The Four-Vortices vector field, introduced by Ferraro et al. (2021), consists of four vortices arranged in a specific pattern. The vector field is defined as:

$$\mathbf{w}(x,y) = s\left(-R_{2,2}(x,y) - R_{4,4}(x,y) - R_{2,5}(x,y) + R_{5,1}(x,y)\right), \tag{28}$$

where each vortex term $R_{a,b}(x,y)$ is given by:

$$R_{a,b}(x,y) = \frac{1}{3\left((x-a)^2+(y-b)^2\right)+1}\begin{bmatrix}-(y-b)\\ x-a\end{bmatrix}, \tag{29}$$

and the scaling factor $s = 1.7$ ensures that the maximum magnitude of the vector field is approximately 1.

The test case involves navigating from point $\mathbf{x}_A = (0,0)$ to point $\mathbf{x}_B = (6,2)$ while minimizing the total travel time. The vessel's speed in calm water (SCW) is constant and set to 1 unit. Ferraro et al. (2021) reported an optimal travel time of $T = 8.95$ units.

Using HADAD, we applied the A* algorithm to generate an initial path on a discretized grid and then refined it using the FMS variational method. We observed that the heuristic weight w in the A* algorithm significantly affects the solution quality and computation time. A higher heuristic weight emphasizes the estimated cost to the goal, guiding the search more directly towards the destination, which can help in finding a local optimum quickly. Conversely, a lower heuristic weight increases exploration of the search space, potentially leading the algorithm to discover a better local optima at the expense of higher computation time.

In our experiments, with a heuristic weight of $w = 0$, HADAD found the best locally optimal path with a travel time of $T = 8.95$, taking 50 s to compute. In an earlier run with $w = 0.5$, the algorithm found a suboptimal path with a travel time of $T = 9.65$ in 40 s. Both results match the locally optimal paths reported by Ferraro et al. (2021).

Fig. 13 illustrates these findings. Sub Fig. 13(a) shows the multiple locally optimal paths identified by Ferraro et al. (2021) (reprinted with permission). Sub Fig. 13(b) presents the solutions obtained by HADAD for different heuristic weights w. This comparison highlights how HADAD can navigate complex optimization landscapes and how the heuristic weight influences the ability to find the global optimum.

6.2. Techy vector field

The Techy vector field, introduced by Techy (2011) and later used by Mannarini and Carelli (2019), is a time-dependent vector field defined as:

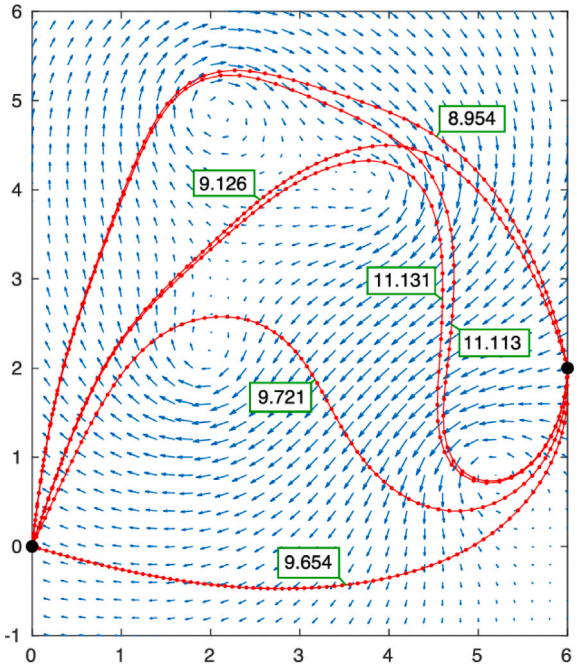
$$\mathbf{w}(x,y,t) = \begin{bmatrix}sx-(t-0.5)y\\ (t-0.5)x+sy\end{bmatrix}, \tag{30}$$

where $s = -0.3$.

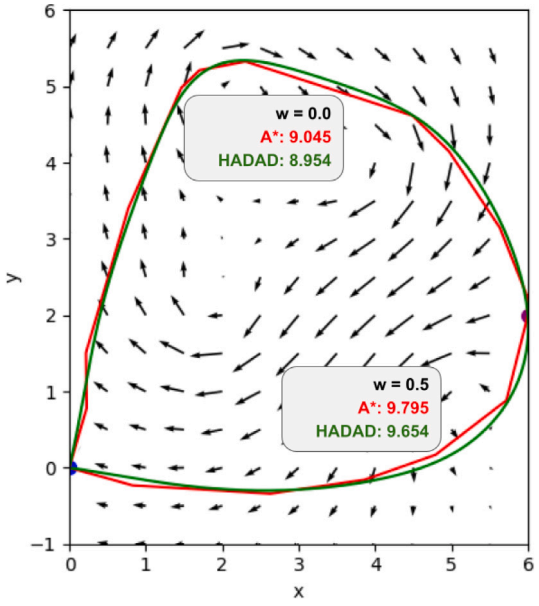
The task is to find the minimum-time path from $\mathbf{x}_A = (\cos(\pi/6), \sin(\pi/6))$ to $\mathbf{x}_B = (0,1)$, with the vessel's speed through water being 1 unit. Note that $\cos(\pi/6) = \sqrt{3}/2 \approx 0.866$ and $\sin(\pi/6) = 1/2 = 0.5$. The time dependence of the vector field introduces additional complexity, which our HADAD algorithm handles effectively.

Table 8
Synthetic vector fields used to validate the HADAD algorithm under different conditions and optimization goals. The table shows the best results found in the literature and the results obtained with HADAD.

Vector Field	Formula	Origin	Destination	Cost	Ref. Cost	HADAD Cost	Comp. Time
Four Vortices (Ferraro et al., 2021)	Eq. (28)	$(0,0)$	$(6,2)$	Time	8.95	8.95	50 s
						9.65	40 s
Techy (Techy, 2011)	Eq. (30)	$(\cos(\pi/6), \sin(\pi/6))$	$(0,1)$	Time	1.03	1.03	1 s
Swirlys (Ferraro et al., 2021)	Eq. (31)	$(0,0)$	$(6,5)$	Fuel	5.73	5.68	80 s



(a) Locally optimal paths showcased in Ferraro et al. [21] (reprinted with permission).



(b) Solutions given by HADAD for different heuristic weights w.

Fig. 13. Four-Vortices vector field and the solutions obtained by Ferraro et al. (2021) and HADAD.

Using HADAD, we computed the optimal path and achieved a travel time of $T = 1.03$ units, matching the reference result reported by Techy (2011). The computation time was approximately 1 s. Fig. 14 shows the optimal trajectory over time.

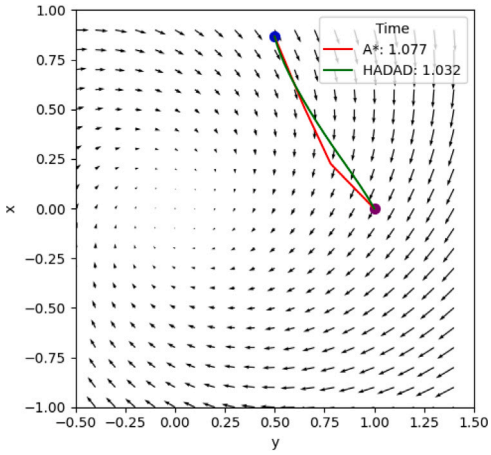


Fig. 14. Optimal time-dependent path in the Techy vector field obtained by HADAD. The vector field is shown at $t = 1.03$.

6.3. Swirlys vector field

The Swirlys vector field, also from Ferraro et al. (2021), presents a complex flow pattern with swirling currents. It is defined as:

$$\mathbf{w}(x, y) = \begin{bmatrix} \cos(2x - y - 6) \\ \frac{2}{3}\sin(y) + x - 3 \end{bmatrix}. \tag{31}$$

The optimization problem involves finding a path from $\mathbf{x}_A = (0,0)$ to $\mathbf{x}_B = (6,5)$ that minimizes a cost function representing fuel consumption, given by:

$$F(\mathbf{x}_{i-1}, \mathbf{x}_i) = \frac{1}{2}\left\| \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{t_i - t_{i-1}} - \mathbf{w}(\mathbf{x}_{i-1}, t_{i-1}) \right\|^2, \tag{32}$$

where $\mathbf{w}(\mathbf{x}, t)$ is the vector field's currents at point $\mathbf{x}$ and time t (although in this example, the vector field is time-independent). This cost function is proportional to the square of the vessel's speed through water (STW), equal to the speed in calm water (SCW) in absence of waves. The optimizer chooses the locations of all $\mathbf{x}_i$ and times t_i, restricted by a total travel time T, which in this example is fixed at 30 units. In the shipping industry, such problems are known as "Just-In-Time" (JIT) arrival planning, where the objective is to minimize fuel consumption while adhering to a fixed time of arrival.

Adjusting HADAD to handle JIT problems requires modifying both the A* and FMS algorithms. The FMS can be adapted by following the approach of Ferraro et al. (2021), which involves redefining the cost function from travel time T to fuel consumption F in the variational formulation. The interested reader is referred to the original study for the full adaptation.

Modifying the A* algorithm to effectively handle JIT problems is more challenging and beyond the scope of this paper. Instead, we made slight adjustments to leverage the exploratory nature of A* and use it as an initial solution for the FMS method, which is well-suited for JIT optimization. Specifically, we implemented the following changes to A*:

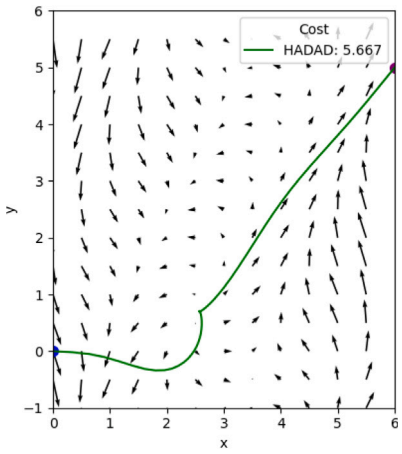


Fig. 15. Optimal path in the Swirlys vector field obtained by HADAD. The path minimizes fuel consumption under a fixed travel time constraint $T = 30$.

1. The timestamp of the destination node D is the fixed time of arrival $t_D = T$.
2. The cost function is redefined as $g(N, P) = F(N, P)$, following Eq. (32), and the heuristic is similarly adjusted to $h(N) = F(N, D)$.
3. To account for variable SCW, we created multiple layers of nodes representing different speeds. We used three discrete speed levels: the original SCW, 0.5 times the original SCW, and 2 times the original SCW. This allows the algorithm to choose nodes with different velocities, effectively increasing the available nodes and enabling variable speed optimization.
4. We introduced a penalty in the heuristic function for any node where the current time $t_N \geq t_D$, to discourage deviations from the fixed time of arrival.
5. No penalties were introduced for $t_N < t_D$, as early arrival times would require higher SCW, thus increasing the cost.

These adjustments allow A* to provide a reasonable initial solution for the JIT-adapted FMS method, despite A*'s limitations in handling variable speeds and fixed arrival times.

Using this JIT-adapted HADAD, we optimized the control inputs and path to minimize the fuel consumption while satisfying the time constraint. Our algorithm achieved a fuel consumption of $F = 5.68$, slightly improving upon the reference value of $F = 5.73$ reported by Ferraro et al. (2021). The computation time for HADAD with this JIT approach was 80 seconds—longer than in the other examples due to the increased number of nodes to explore because of variable speeds. Fig. 15 illustrates the optimal path obtained. Note that the optimal trajectory is bound to arrive in $T = 30$ and needs to find a place to spend some extra time. To minimize fuel consumption (which in this model is proportional to the square of SCW), the trajectory finds a point with weaker currents to add some extra time. While this situation is just a toy example and far from real applications, it highlights the soundness and coherence of JIT approach with HADAD.

7. Discussion

To test the efficiency of HADAD, we developed a benchmark with 1,560 different problem instances. This benchmark is available on our website.[2] It offers easily accessible weather data files for download and includes auxiliary functions for interpolation of weather data, speed reduction models, and scoring functions for other researchers willing to tackle simplified weather routing optimization problems and score their proposed algorithms against the benchmark.

Our benchmark is composed of five ODPs in different oceans connecting major hubs, as well as weather data covering a full year, thus allowing for both spatial and temporal variability. The optimization problem for this initial version of the benchmark involves minimizing travel time at a fixed speed over water. The vessel speed through water is affected by ocean waves and currents according to a simple model, for which we provide the relevant code and explanations.

During our testings on this benchmark, when sailing at low speeds we observe consistent gains compared to the shortest distance route. These gains are mainly due to the smart use of ocean currents, as shown in Table 7, and usually range between $2\%-6\%$. However, at higher vessel speeds where environmental conditions have minimal impact, the HADAD algorithm occasionally produces routes with negative gains, i.e. routes that take longer time than the minimum distance. Upon investigation, we identified that this issue arises due to the inherent limitations of the grid structure used in the A* component of our algorithm. The discretization of the search space prevents the algorithm from perfectly replicating the optimal straight-line path in a continuous plane, leading to suboptimal routes in these cases. This limitation highlights an area where the current version of HADAD can be improved. Future work will focus on refining the grid representation, possibly through increased resolution or adaptive grids, and incorporating more complex cost functions that can mitigate the effects of discretization at higher speeds.

Regarding the benchmark construction, an area of improvement is its dynamic models. Beyond ship designs, a better understanding and simulation of friction and resistance terms are crucial for accurate power prediction. Currently, we only consider the vessel's length, displacement, and block coefficient. However, other variables impact performance, such as beam, draft, prismatic coefficient, midships coefficient factor, presence and design of the bulbous bow, and longitudinal center of buoyancy (Holtrop et al., 1982; Faltinsen, 1980). All these factors are necessary to simulate and compute resistances, such as appendage resistance, wave-making and wave-breaking resistance, and bulbous bow pressure near the water surface (Holtrop et al., 1982). These resistance terms would inform future ship designs and refitting and offer the opportunity to use main engine power output or fuel consumption, directly influencing fuel and operational costs, as a viable cost function instead of time spent on route. This aligns with sustainable development strategies set by the IMO and the UN (IMO, 2017, 2023).

Another particular challenge faced by weather routing is quantifying uncertainty in weather forecasts (Wang et al., 2020). In this study, we assume that ocean forecasts are available for the entire journey duration, as our benchmark is based on past reanalysis data. However, in real conditions, forecasts are only available up to ten days into the future (Copernicus, 2024), with decreasing accuracy further ahead. Thus, uncertainties in weather forecasts demand robust algorithms and benchmarks that are able to quantify and handle the uncertainty inherent in weather prediction.

The validation results on synthetic vector fields demonstrate that HADAD is capable of finding optimal or near-optimal solutions in various synthetic vector fields, matching or slightly improving upon reference results from the literature. The combination of the A* graph search for global exploration and the FMS variational method for local refinement allows HADAD to effectively handle complex flow patterns and time-dependent fields. For instance, in the Swirlys vector field (Ferraro et al., 2021), HADAD achieved slightly lower fuel consumption than the reference method, despite both using FMS for refinement. This improvement is attributed to the exploratory advantage provided by the A* component, suggesting that HADAD can potentially find better solutions by leveraging its combination of exploration and refinement.

[2] http://benchmark.weathernavigation.com/

These validation tests confirm the robustness and reliability of HADAD in solving navigation problems under varying conditions and cost functions. The ability to match or surpass reference results in controlled environments offers confidence in the algorithm's potential to perform effectively in real-world scenarios.

8. Conclusion

This study introduced HADAD, a new optimization method for ship weather routing. HADAD is a combined algorithm that first utilizes a A* graph search on a hexagonal grid, tiling the Earth and finding a route of minimum time on a discretized space. The method then moves to a continuous space and refines the route using a variational gradient descent algorithm named FMS. Not only does FMS provide a smoother trajectory, but it is also guaranteed to converge to a locally optimal solution of the variational problem, as proved in Ferraro et al. (2022). The shortcoming of using FMS alone is that this locally optimal trajectory might be far from the global optimum, as it happens with many gradient descent methods. To address this issue, Precioso et al. (2024) implemented an exploration phase prior to FMS, using a shooting method that evolves many time-optimal trajectories in parallel, then taking the best candidate as the initial solution for the FMS. Our proposed algorithm is a concatenation of A* and FMS, where A* plays the role of a global exploration, while FMS behaves as a local greedy search that is guaranteed to converge to a local minimum. We have named this new algorithm HADAD.

Under these conditions, HADAD showed strong adaptability, solving routing problems without being affected by land density. This marks a clear improvement over earlier algorithms that used FMS refinement (Precioso et al., 2024). At slow steaming speeds (12 knots), HADAD reduced travel time by up to 5% compared to the shortest distance route. The algorithm can also easily adapt to other objectives, such as fuel consumption or emissions, which we will explore in future work.

In terms of computational efficiency, HADAD usually finds a local optimum in about two minutes, comparable to the state of the art in routing algorithms as shown in Table 1. However, performance decreases for long routes, which we aim to improve with better heuristics or parallelization in future work.

Despite these promising results, it is important to acknowledge the limitations of the current version of HADAD. Specifically, the algorithm's performance can be hindered in scenarios involving high vessel speeds where environmental factors have negligible influence. The grid-based discretization of the A* search space can prevent HADAD from identifying the optimal path, leading to minor negative gains compared to the shortest distance route. However, these instances are rare – occurring in less than 1% of scenarios – and the losses are minimal, below 0.4%. Addressing these limitations will involve increasing grid resolution or refining cost functions, though these adjustments may require additional computational resources.

Future work will also focus on adapting A* for Just-In-Time (JIT) arrival planning, a crucial aspect of modern maritime transportation that aims to optimize fuel consumption while ensuring precise arrival times. Implementing JIT planning will enhance HADAD's applicability by enabling dynamic operation planning and better coordination between ports and ships, thereby minimizing unnecessary fuel use and resource consumption. This development aligns with industry initiatives such as the DYNAPORT (European Union, 2024a) and MISSION (European Union, 2024b) projects, which seek to develop optimization, communication and coordination tools for Just-in-Time arrival, to reduce ship fuel consumption and increase port efficiency.

In conclusion, HADAD offers a robust and efficient solution for ship weather routing by effectively combining global exploration with local refinement. It achieves a balance between computational efficiency and route optimality, demonstrating notable reductions in travel time under various conditions. By addressing current limitations and expanding its capabilities, HADAD holds significant promise for practical applications in maritime navigation and contributes valuable advancements to the field of weather routing optimization.

CRediT authorship contribution statement

Javier Jiménez de la Jara: Writing – original draft, Software, Data curation. **Daniel Precioso:** Writing – original draft, Project administration. **Louis Bu:** Writing – original draft, Validation, Investigation. **M. Victoria Redondo-Neble:** Resources. **Robert Milson:** Supervision. **Rafael Ballester-Ripoll:** Validation, Supervision. **David Gómez-Ullate:** Writing – review & editing, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This publication is part of the project "Optimization of maritime routes for a more efficient, safer and decarbonized maritime transport", which is funded by the BBVA Foundation and Grant TED2021-129455B-I00 from MCIN/AEI/ 10.13039/501100011033 and the European Union "NextGenerationEU/PRTR".

We also acknowledge support from the project PID2021-122154NB-I00, funded by MICIU/AEI/10.13039/501100011033 and "ERDF A Way of making Europe". The authors would like to thank the MITACS International Accelerate program that enabled the visit of LB to Madrid in the summer of 2024, where the work was completed.

Appendix A. Expanded tables

Table 9 complements Table 1 with additional information about recent studies in weather routing.

Appendix B. Weather data

B.1. Ocean General Circulation Models (OGCMs)

Ocean General Circulation Models (OGCMs) models simulate the physical interactions of the world's oceans and atmospheres, capturing the dynamics of ocean currents, temperature, salinity, and ice cover over time (McWilliams, 2000). By mathematically modeling the fundamental laws of fluid dynamics and thermodynamics, OGCMs can predict changes in the oceanic circulation and its interaction with the atmosphere, land, and sea ice. OGCMs can be classified based on the grid type used to distribute the data. They combine two grids: a horizontal grid covering the Earth surface and a vertical one covering the depth levels of the ocean or height levels of the atmosphere. The most commonly used horizontal grids include:

1. **Finite Differences:** These are the most used grids, and they discretize the space evenly by fixed distance or degrees of arc, spanning both the longitude and latitude axes. The distribution of data usually follows Arakawa's structure (Arakawa and Lamb, 1977) that has five types (A to E) ordered by increasing complexity and accuracy.
2. **Finite Element:** This is the second most popular choice for ocean modeling, especially for coastlines and offshore regions. In a finite element grid, weather variables are discretized by triangular regions. It can adjust the sizes of triangles according to the weather variable, or the complexity of the shoreline due to its 'fractal' nature. This dynamic nature offers great flexibility and ease to adjust the detail of the model where it is needed most (Wang et al., 2014).

Table 9
Examples of recent weather routing studies, indicating the algorithms they used, which algorithms they compared against, their cost function and whether the data and code is publicly online.

Reference	Compared Against	Cost Function	Open-Source?
VISIR-2 (Mannarini et al., 2024)	Dijkstra's graph search	CO_2 emissions	Yes
Hybrid Search (Precioso et al., 2024)	A* graph search	Travel time	No
PRM (Charalambopoulos et al., 2023)	Dijkstra's graph search	Travel time and fuel consumption	No
OSN (Huang et al., 2023)	Graph search	Travel distance	No
w-MOEA/D (Szlapczynski et al., 2023)	Isochrones	Travel time, fuel consumption and safety	No
WRM (Grandcolas, 2022)	Itself	Fuel consumption	No
SIMROUTE (Grifoll et al., 2022)	Dijkstra's graph search	Travel time	Yes
HNDS-MPSO (Zhao et al., 2022)	A* graph search	Travel time, fuel consumption and safety	No
Kuhlemann and Tierney (2020)	Graph search	Fuel consumption	No
Tsai et al. (2021)	Real data	Travel time, fuel consumption and distance	No
Wang et al. (2021)	Dijkstra's graph search	Fuel consumption and GHG emissions	No
Gkerekos and Lazakis (2020)	Real data	Fuel consumption	No
Vettor et al. (2020)	A* graph search	Travel time, fuel consumption and safety	No

3. **Spectral:** These are the least used grids in ocean models, due to the difficulties caused by land boundaries, but they are widely used in atmospheric studies (Polvani et al., 2004).

There are various vertical discretization methods that accurately model the depth of the ocean, the most common ones being Z-coordinates, Sigma(S)-coordinates, and Isopycnal coordinates. All of these methods offers a great amount of accuracy on the surface of the ocean, which is the region relevant for this study.

B.2. Nucleus for European Modelling of the Ocean (NEMO)

The Nucleus for European Modelling of the Ocean (Madec et al., 2017) represents an important component for the global oceanographic research. Developed through a collaborative effort by several leading European research institutions, it is a versatile modeling framework designed to study the ocean and its interactions with the lower atmosphere, sea ice, and biogeochemical processes.

Referring to the grid system, NEMO employs an orthogonal curvilinear grid for horizontal representation and combines Z and S coordinates for the vertical dimension, and distribution of output variables is arranged in a three-dimensional Arakawa C-type grid (Arakawa and Lamb, 1977).

NEMO's framework offers a holistic approach to ocean modeling by integrating multiple key components. It simulates oceanic physical processes using advanced numerical methods to understand currents, temperature, salinity, and sea level changes (Madec et al., 2017). Additionally, NEMO handles the complex interactions between the ocean and sea ice, including formation and melting, and their effects on circulation (Vancoppenolle et al., 2023). It also explores marine ecosystems and bio-chemical cycles to study the ocean's contribution to the carbon cycle and environmental responses (NEMO TOP Working Group, 2022).

B.3. Météo-France Wave Model (MFWAM)

The Météo-France Wave Model (MFWAM) is an advanced, flexible modeling framework designed to simulate the generation, propagation, and dissipation of ocean waves. It serves both as a scientific tool for understanding wave dynamics and as a core element in operational wave forecasting systems.

MFWAM employs the ECWAM-IFS-38R2 (European Centre for Medium-Range Weather Forecasts, 2016b) computing code, incorporating dissipation terms developed by Ardhuin et al. (2010). This foundation ensures robust performance in simulating wave dynamics. In November 2014, the MFWAM model received significant upgrades, thanks to advancements from the European research project "My Wave" (Janssen et al., 2016; E.U. Copernicus Marine Service Information (CMEMS), 2024).

Operationally, the MFWAM model is driven by 6-hourly analysis and 3-hourly forecast winds from the ECMWF-IFS atmospheric system. The wave spectrum is discretized into 24 directions and 30 frequencies, ranging from 0.035 Hz to 0.58 Hz, providing detailed wave information. Additionally, it utilizes partitioning to separate the swell spectrum into primary and secondary swells, allowing for more nuanced and precise wave predictions.

B.4. Data processing

The maintenance of waves (MFWAM) and currents (NEMO) data are undertaken by Copernicus (2024), a program initiated by the European Union designed to enhance European informational services through the utilization of satellite Earth Observation and in situ (non-space) data.

The primary aim of Copernicus is to provide comprehensive monitoring and forecasting of the environmental state across terrestrial, marine, and atmospheric domains. This project supports a broad spectrum of objectives, including aiding climate change mitigation and adaptation strategies, fostering efficient emergency management practices, and enhancing the security and well-being of European citizens.

`GLOBAL_ANALYSISFORECAST_PHY_001_024` is a Copernicus product that provides a comprehensive dataset including over 30 variables such as salinity, potential temperature, and currents, among others (E.U. Copernicus Marine Service Information (CMEMS), 2024). Within this project, the focus is primarily on the current data, which is analyzed in terms of its vertical ($w_{\perp}$) and horizontal ($w_{\parallel}$) components. The dataset is structured on a regular grid with a resolution of 1/12°, spanning from 180°W to 179.92°E and 89°S to 90°N (4320 × 2041 resolution). This product features 50 depth levels, arranged on an Arakawa C type grid (E.U. Copernicus Marine Service Information (CMEMS), 2024). However, only the surface level is utilized for this analysis. Considering that the usual draft of a container vessel is 12 m (Rinauro et al., 2024), the currents vary 0.01 m/s on average between the surface layer and at 12 m depth, which has a negligible effect compared with a vessel's typical speed. We further discuss this study in Appendix B.5.

`GLOBAL_ANALYSISFORECAST_WAV_001_027` is another product offered by Copernicus, which provides users the sea surface significant wave height and direction, along with a comprehensive list of variables (E.U. Copernicus Marine Service Information (CMEMS), 2024). The spatial resolution of this dataset is 1/12°, spanning 180°W to 179.92°E and 89°S to 90°N, the same data dimensions as the product mentioned previously; however the temporal resolution is 3 h.

In this model, currents are recorded on a daily basis due to their relatively stable nature, with significant changes occurring over longer periods, and waves are stored every 3 h. Consequently, the dataset comprises a NetCDF (Network Common Data Form) file for each day of the year, adhering to the naming convention: `YYYY-MM-DD.nc`. NetCDF is a set of software libraries and machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data (Unidata Program Center, 2023). This arrangement results in a collection of 365 files, each with near 9 million data points for each variable, amounting to 24 GBs of currents data and 97 GB of waves data.

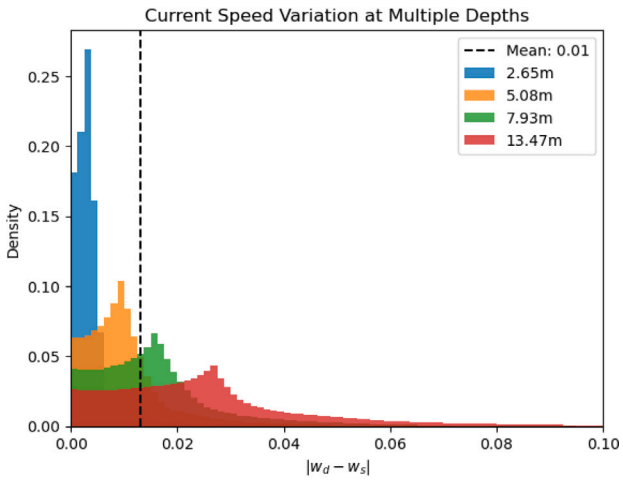


Fig. 16. Difference in currents between the surface (w_s) and various depths (w_d). Only a selection of the depths used to calculate the mean has been plotted to improve and simplify visualization.

B.5. Currents variation in depth

Given that the typical draft of a container vessel is around 12 m, a study was conducted using current velocity data at various depths over the whole oceans on a given day. The results show that the difference in absolute value between ocean currents at the surface and their values at different depths are on average 0.01 m/s, which is negligible in comparison with typical vessel speeds. As illustrated in Fig. 16, the differences are larger at larger depths, but even at 13 m this difference is typically below 0.05 m/s. This justifies our choice to consider only ocean current speed on the surface, regardless of the vessel's draft.

Appendix C. Time complexity study

The HADAD algorithm is composed of two distinct phases, A* search and FMS refinement, making it necessary to analyze each phase separately. The first phase involves A*, which can have a complexity close to that of Dijkstra's algorithm in the worst case. As defined in Barbehenn (1998), its complexity is

$$O(|E| + |V| \log(|V|)) \tag{33}$$

where $\boldsymbol{E}$ is the set of edges and $\boldsymbol{V}$ is the set of vertices. In our case, the graph is a bidirectional grid, so the number of edges is simply a multiple of the number of vertices (***K*-order neighbors**). Consequently, the complexity simplifies to

$$O(|V| + |V| \log(|V|)) = O(|V| \log(|V|)) \tag{34}$$

using the sum rule.

On the other hand, to analyze the complexity of the FMS, it is necessary to consider the behavior of the variational algorithm. FMS performs a transformation on each way-point (num_waypoints) along the path, and this is repeated for a user-defined number of iterations (num_iter), resulting in a complexity of:

$$O(\text{num\_iter} \cdot \text{num\_waypoints}) \tag{35}$$

Since the number of iterations remains constant regardless of the number of cells ($|V|$), and assuming the path length is mostly proportional to $|V|$. By the multiplication rule, this can be simplified to $\mathbf{O}(|\mathbf{V}|)$.

In conclusion, the overall complexity of the HADAD algorithm is defined as:

$$O(|V| \log |V| + |V|) \approx O(|V| \log |V|) \tag{36}$$

Appendix D. Interpolation

Given the dynamic nature of vessel locations at sea, it is imperative to estimate the ocean current magnitudes at these positions accurately. To achieve this, we employ bicubic interpolation on meteorological datasets, including ocean currents speed in northward and eastward directions, and wave height and northward direction.

Let $s_{i,j}$ where $i = 0, 1, \ldots, 2147$ and $j = 0, 1, \ldots, 4319$, represent any meteorological condition, indexed by the grid point along the latitudinal and longitudinal coordinates. The (i, j) index are determined by the grid solution/netCDF file dimensions of Copernicus product (Copernicus, 2024). We construct a bicubic polynomial like the following:

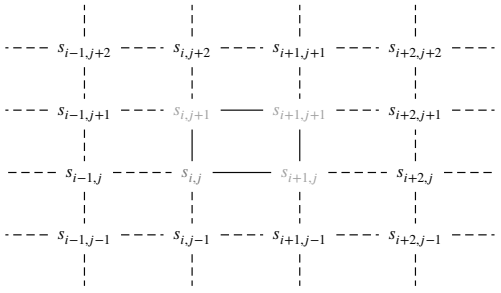
We build a bicubic polynomial for each $1/12° \times 1/12°$ square, this polynomial can be represented as:

$$f_{(i,j)}(x, y) = \sum_{m=0}^{3} \sum_{n=0}^{3} a_{i,j} \cdot x^m \cdot y^n$$

In matrix form, it can also be expressed as:

$$f_{(i,j)}(x, y) = \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} 1 \\ y \\ y^2 \\ y^3 \end{bmatrix}$$

To compute the coefficients $a_{i,j}$, it is necessary to have a minimum of 16 linearly independent equations. These equations can be obtained by initially considering the immediate 4×4 square that surrounds the grid point of interest, indexed as (i, j) (highlighted in red in the diagram below). Subsequently, the values at these 16 points are taken into account, as depicted in the following diagram:



We arrange the above values into a length 16 vector $\vec{s}$, and we define the interpolation matrix $\mathbf{M}$ as a block matrix like the following:

$$\mathbf{M} = \begin{bmatrix} \mathbf{0} & 2\mathbf{A} & \mathbf{0} & \mathbf{0} \\ 2\mathbf{B} & -\mathbf{A} & 2\mathbf{A} & \mathbf{B} \\ \mathbf{A} & -2\mathbf{A} & \mathbf{A} & \mathbf{0} \\ \mathbf{B} & \mathbf{A} & -\mathbf{A} & -\mathbf{B} \end{bmatrix} \tag{37}$$

where

$$\mathbf{A} = \frac{1}{3} \begin{bmatrix} 0 & 6 & 0 & 0 \\ -2 & -3 & 6 & -2 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \tag{38}$$

and

$$\mathbf{B} = \frac{1}{9} \begin{bmatrix} 0 & -6 & 0 & 0 \\ 2 & 3 & -6 & 1 \\ -3 & 6 & -3 & 0 \\ 1 & -3 & 3 & -1 \end{bmatrix} \tag{39}$$

As these matrix coefficients are defined, we also ensure that these polynomials agrees on the boundaries, i.e.,

$$\begin{cases} f_{(i,j)}(-, 0) = f_{(i,j-1)}(-, 1) \\ f_{(i,j)}(-, 1) = f_{(i,j+1)}(-, 0) \end{cases} \tag{40}$$

$$\begin{cases} f_{(i,j)}(0, -) = f_{(i-1,j)}(1, -) \\ f_{(i,j)}(1, -) = f_{(i+1,j)}(0, -) \end{cases} \tag{41}$$

to ensure that no sudden discontinuity is introduced in interpolation.

Then coefficients vector $\vec{a}$ can be determined by multiplying $\vec{s}$ with a 16×16 matrix $\mathbf{M}$ as follows:

$$\mathbf{M}\vec{s} = \vec{a}$$

Where s and a are defined as follows:

$$\vec{s} = \begin{bmatrix} s_{i-1,j+2} \\ s_{i,j+2} \\ s_{i+1,j+1} \\ s_{i+2,j+2} \\ s_{i-1,j+1} \\ s_{i,j+1} \\ s_{i+1,j+1} \\ s_{i+2,j+1} \\ s_{i-1,j} \\ s_{i,j} \\ s_{i+1,j} \\ s_{i+2,j} \\ s_{i-1,j-1} \\ s_{i,j-1} \\ s_{i+1,j-1} \\ s_{i+2,j-1} \end{bmatrix} \vec{a}_{(i,j)} = \begin{bmatrix} a_{00} \\ a_{01} \\ a_{02} \\ a_{03} \\ \hline a_{10} \\ a_{11} \\ a_{12} \\ a_{13} \\ \hline a_{20} \\ a_{21} \\ a_{22} \\ a_{23} \\ \hline a_{30} \\ a_{31} \\ a_{32} \\ a_{33} \end{bmatrix} \tag{42}$$

Note that these coefficients $\vec{a}$, is only valid within the square with corners $s_{i,j}, s_{i+1,j}, s_{i+1,j+1}, s_{i,j+1}$.

It is evident that the resulting vector $\vec{a}$ is also of length 16. To facilitate further analysis, we reshape this vector into a 4×4 matrix denoted as $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix}$$

The above matrix of 16 coefficients are dependent on the grid point of interest (i, j). Using this matrix, we can construct the bicubic polynomial $f_{(i,j)}(x, y)$ at any point (x, y) in the square.

This process is very similar to convolution in signal and image processing, which we can utilize `Numpy`'s strengths in multi-dimension matrix algebra and computing to interpolate the entire data grid all at once instead of a nested for-loop, which drastically decreases compute time.

Appendix E. A⋆ parameter grid search

We have run experiments for a number of different configurations for the A⋆ algorithm, namely

- Grid resolution (H3 library): 3, 4, 5
- K-order neighbors: 1, 2, 3
- Weight of the heuristic w: 0.5, 1.0, 1.25

This amounts to 27 different configurations, run across all ten ODPs (five pairs of ports in both directions) at three different velocities (6, 12 and 24 knots), for a total of 810 different experiments. All departure dates were set to the 1st of January of 2023. For each experiment we registered the computation time of the A⋆ algorithm and the time it takes the vessel to reach its destination — the target to be optimized. We then compared that travel time with the time taken by following the minimum distance route (orthodromic navigation), computing the percentage gain of the A⋆ with respect to it. All experiments where conducted on the same machine.[3]

It is worth noting that A⋆ was not able to find a route for 90 out of the 810 experiments. The main source of this issue were the ODPs between EGHRG and MYKUL. Due to the narrow Suez canal, some grid configurations did not present any feasible connection between the start and end nodes. Particularly, grid resolution 4 with 1st-order neighbors and grid resolution 3 with 1st- and 2nd-order did not reach a feasible solution. This challenge is not a limitation of the A⋆ algorithm itself but rather a consequence of the H3 graph structure's resolution constraints. To address this, one potential improvement is to implement an adaptive multi-scale resolution strategy that increases grid granularity in the presence of complex land masses like archipelagos or straits.

[3] Intel Core i9-14900K, with 128GiB RAM.

Table 10
Pearson correlation coefficient (PCC) between the gains produced by A⋆ (compared with the minimum distance), its computation time and the different parameters of this algorithm.

Configuration Parameters	Gain	Compute time
Vessel speed	−0.258	0.030
K-order neighbors	0.678	0.146
Grid resolution	0.211	0.269
Weight of the heuristic	−0.253	−0.242
Explored nodes	0.150	0.757
Wave height	0.107	0.030

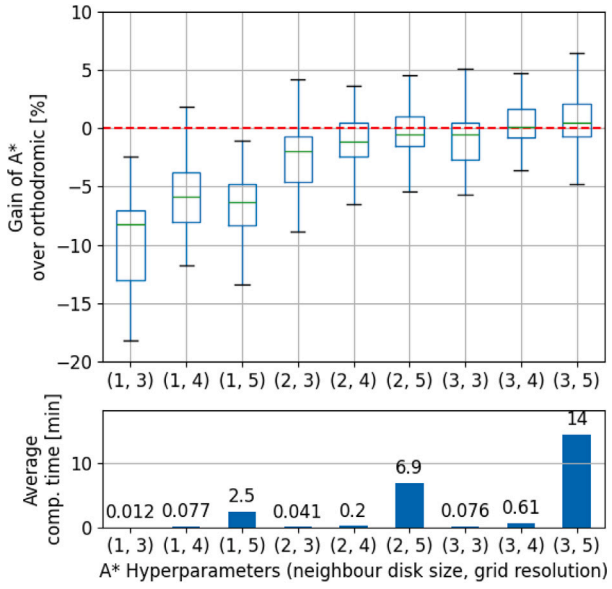


Fig. 17. Parameter settings for A⋆, grouped by K-order neighbors and grid resolution as defined by the H3 library (Uber Technologies, Inc., 2018). The gain of A⋆ over orthodromic navigation is shown, and computation times are plotted below.

We computed the Pearson correlation (Freedman et al., 2007) between parameters and results, to better understand their impact. These Pearson coefficients are shown in Table 10, comparing instance's parameters such as speed, explored nodes, and wave height. In relation to problem instances, it is evident that higher vessel speeds result in less gains. On the other hand, an increased number of nodes and the presence of strong currents and high waves increase the potential gains achieved by A⋆. Among A⋆ parameters, increasing the K-order neighbors significantly improves optimization, because it adds more nodes to explore at each step. Additionally, the weight of the heuristic and grid resolution greatly impact computation time, which is crucial for deployment and implementation of this system.

Fig. 17 groups the A⋆ gains by K-order neighbors and grid resolution, validating our expectations: a finer grid resolution and a bigger neighbor order opens more paths to explore, and thus improves the overall results. We also observe, however, that a grid resolution of 5 increases the computation time by at least an order of magnitude while only managing to net gains similar to resolution 4. To balance gains with a reasonable computation time, we will choose a grid resolution of 4 with 3rd-order neighbors. Next we decide a weight for the heuristic.

Table 11
Gain and computation time of A*, showing mean (std) across configurations. Grid resolution is fixed to 4, and 3rd-grade neighbors.

Speed (knots)	Weight of Heuristic	Gain (%)	Compute time (min)
6	0.5	3.79 (2.43)	1.51 (1.73)
6	1.0	3.00 (1.83)	0.36 (0.65)
6	1.25	−0.56 (2.46)	0.14 (0.23)
12	0.5	1.34 (2.03)	1.03 (1.06)
12	1.0	0.88 (1.72)	0.37 (0.46)
12	1.25	−1.89 (1.74)	0.09 (0.17)
24	0.5	0.01 (0.84)	1.48 (1.82)
24	1.0	−0.16 (0.88)	0.42 (0.52)
24	1.25	−2.75 (1.41)	0.11 (0.21)

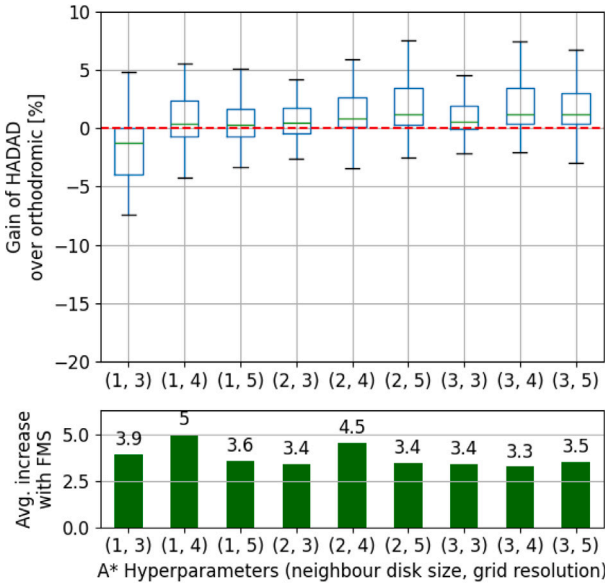


Fig. 18. The box-plot show the gain of HADAD over orthodromic navigation, for every A* configuration. Below them, the bars show the average increase (in gain %) that FMS achieves when applied to A*.

Looking at Table 11 we conclude that an heuristic weight of 0.5 offers the best results without a significant cost in computation time. We also observe, however, that A* still struggles to achieve gains over the minimum distance routes in some scenarios. To improve its results, we will apply a FMS refinement discussed in Section 3.2 to every A* output. This ensures that FMS will always output a solution at least as good as the seed route provided by A*.

FMS is able to improve any A* configuration greatly, as illustrated in Fig. 18. FMS is able to neutralize the disadvantages resulted by sub-optimal A* configurations, to the extent where the average gain in each A* configuration is always greater than 0. As such, worse A* configurations benefit more from FMS, witnessing an increase of around 8% in gains for 1st-grade neighbors. The choice of A* parameters is not so crucial after applying the FMS algorithm, whose addition is a great improvement over a pure graph optimization method.

References

Alderson, D.L., Funk, D., Gera, R., 2020. Analysis of the global maritime transportation system as a layered network. J. Transp. Secur. 13, 291–325.

Álvarez, N.G., Adenso-Díaz, B., Calzada-Infante, L., 2021. Maritime traffic as a complex network: A systematic review. Netw. Spat. Econ. 21, 387–417.

Arakawa, A., Lamb, V.R., 1977. Computational design of the basic dynamical processes of the UCLA general circulation model. Methods Comput. Phys.: Adv. Res. Appl. 17, 173–265. http://dx.doi.org/10.1016/B978-0-12-460817-7.50009-4.

Ardhuin, F., Magne, R., Filipot, J.-F., Van, A., Westhuysen, D., Roland, A., Queffeulou, P., Lefevre, J.-M., Aouf, L., Babanin, A., Collard, F., 2010. Semi-empirical dissipation source functions for wind-wave models: part I, definition, calibration and validation at global scales. J. Phys. Oceanogr. 40.

Barbehenn, M., 1998. A note on the complexity of Dijkstra's algorithm for graphs with weighted vertices. IEEE Trans. Comput. 47, 263. http://dx.doi.org/10.1109/12.663776.

Buduma, N., Locascio, N., 2017. Fundamentals of deep learning: Designing next-generation machine intelligence algorithms. O'Reilly Media, URL: https://books.google.es/books?id=SL0BvgAACAAJ.

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.

Chen, Y., Tian, W., Mao, W., 2024. Strategies to improve the isochrone algorithm for ship voyage optimisation. Ships Offshore Struct. 1–13.

Copernicus, 2024. Europe's eyes on earth. URL: https://www.copernicus.eu/es. (Accessed 10 July 2024).

Dijkstra, E.W., et al., 1959. A note on two problems in connexion with graphs. Numer. Math. 1, 269–271.

Ducruet, C., 2020. The geography of maritime networks: A critical review. J. Transp. Geogr. 88, 102824. http://dx.doi.org/10.1016/j.jtrangeo.2020.102824.

Ebendt, R., Drechsler, R., 2009. Weighted A* search - unifying view and application. Artificial Intelligence 173, 1310–1342. http://dx.doi.org/10.1016/J.ARTINT.2009.06.004.

E.U. Copernicus Marine Service Information (CMEMS), 2024. GLOBAL_ANALYSISFORECAST_PHY_001_024: Ocean/Sea Physical Analysis and Forecast. http://dx.doi.org/10.48670/moi-00016, https://data.marine.copernicus.eu/product/GLOBAL_ANALYSISFORECAST_PHY_001_024/description. (Accessed 10 July 2024).

E.U. Copernicus Marine Service Information (CMEMS), 2024. GLOBAL_ANALYSISFORECAST_WAV_001_027: Global Ocean Waves Analysis and Forecast. http://dx.doi.org/10.48670/moi-00017, https://data.marine.copernicus.eu/product/GLOBAL_ANALYSISFORECAST_WAV_001_027/description. (Accessed 10 July 2024).

European Centre for Medium-Range Weather Forecasts, 2016. ECMWF IFS CY41r2 high-resolution operational forecasts. http://dx.doi.org/10.5065/D68050ZV.

European Centre for Medium-Range Weather Forecasts, 2016b. Part VII: ECMWF Wave Model. URL: https://www.ecmwf.int/sites/default/files/elibrary/2016/17120-part-vii-ecmwf-wave-model.pdf. (Accessed 10 July 2024).

European Union, 2024a. DYNAPORT – DYnamic NAvigation and Port call Optimisation in Real Time. URL: https://dynaport.eu/.

European Union, 2024b. MISSION Project. URL: https://missionproject.eu/.

Faltinsen, O.M., 1980. Prediction of resistance and propulsion of a ship in a seaway. In: 13th Symposium on Naval Hydrodynamics, Tokyo. pp. 505–529.

Ferraro, S.J., de Diego, D.M., de Almagro, R.T.S.M., 2021. Parallel iterative methods for variational integration applied to navigation problems. IFAC-PapersOnLine 54, 321–326. http://dx.doi.org/10.1016/J.IFACOL.2021.11.097.

Ferraro, S.J., Martín de Diego, D., Sato Martín de Almagro, R.T., 2022. A parallel iterative method for variational integration. arXiv preprint arXiv:2206.08968.

Frasinaru, C., Raschip, M., 2019. Greedy best-first search for the optimal-size sorting network problem. Procedia Comput. Sci. 159, 447–454. http://dx.doi.org/10.1016/J.PROCS.2019.09.199.

Freedman, D., Pisani, R., Purves, R., 2007. In: Pisani, R., Purves (Eds.), Statistics (international student edition), 4th edn. WW Norton & Company, New York.

Ge, J., Zhang, Q., Wan, Z., et al., 2022. Regional operating patterns of world container shipping network: A perspective from motif identification. Phys. A 607, 128171.

Gkerekos, C., Lazakis, I., 2020. A novel, data-driven heuristic framework for vessel weather routing. Ocean Eng. 197, 106887.

Gleeson, E., Clancy, C., Zubiate, L., Janjić, J., Gallagher, S., Dias, F., 2019. Teleconnections and extreme ocean states in the Northeast Atlantic Ocean. Adv. Sci. Res. 16, 11–29. http://dx.doi.org/10.5194/asr-16-11-2019, URL: https://asr.copernicus.org/articles/16/11/2019/.

Grandcolas, S., 2022. A metaheuristic algorithm for ship weather routing. In: Operations Research Forum. Oper. Res. 3, 35.

Grifoll, M., Borén, C., Castells-Sanabra, M., 2022. A comprehensive ship weather routing system using CMEMS products and A* algorithm. Ocean Eng. 255, 111427.

Hart, P.E., Nilsson, N.J., Raphael, B., 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Trans. Syst. Sci. Cybern. 4, 100–107. http://dx.doi.org/10.1109/TSSC.1968.300136.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., Steinhardt, J., 2021. Measuring massive multitask language understanding. arXiv:2009.03300.

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., Simmons, A., Soci, C., Abdalla, S., Abellan, X., Balsamo, G., Bechtold, P., Biavati, G., Bidlot, J., Bonavita, M., De Chiara, G., Dahlgren, P., Dee, D., Diamantakis, M., Dragani, R., Flemming, J., Forbes, R., Fuentes, M., Geer, A., Haimberger, L., Healy, S., Hogan, R.J., Hólm, E., Janisková, M., Keeley, S., Laloyaux, P., Lopez, P., Lupu, C., Radnoti, G., de Rosnay, P., Rozum, I., Vamborg, F., Villaume, S., Thépaut, J.-N., 2020. The ERA5 global reanalysis. Q. J. R. Meteorol. Soc. 146, 1999–2049. http://dx.doi.org/10.1002/qj.3803.

Holtrop, J., Mennen, G., et al., 1982. An approximate power prediction method. Int. Shipbuild. Prog. 29, 166–170.
Huang, Y., Xiao, Y., Wang, H., Yi, H., 2023. A rapid globe-wide shortest route planning algorithm based on two-layer oceanic shortcut network considering great circle distance. Ocean Eng. 287, 115761.
IMO, 2017. IMO and sustainable development. URL: https://wwwcdn.imo.org/localresources/en/MediaCentre/HotTopics/Documents/IMO%20SDG%20Brochure.pdf.
IMO, 2020a. Fourth IMO GHG Study 2020 - doc. MEPC59/INF.10. International Maritime Organization (IMO), London, UK.
IMO, 2020b. Just In Time Arrival Guide: Barriers and Potential Solutions. International Maritime Organization (IMO), London, UK.
IMO, 2023. Development of the strategic plan for 2024 to 2029. URL: https://wwwcdn.imo.org/localresources/en/About/strategy/Documents/A%2033-Res.1173.pdf.
Janssen, P., Aouf, L., Behrens, A., Korres, G., Cavalieri, L., Christiensen, K., Breivik, O., 2016. MyWave Project. URL: https://www.ecmwf.int/en/research/projects/mywave. (Accessed 10 July 2024).
Kim, M., Hizir, O., Turan, O., Day, S., Incecik, A., 2017. Estimation of added resistance and ship speed loss in a seaway. Ocean Eng. 141, 465–476.
Kim, Y.-R., Steen, S., 2022. Application of machine learning algorithms for predicting added resistance in arbitrary wave headings of a ship. In: Volume 5B: Ocean Engineering; Honoring Symposium for Professor Günther F. Clauss on Hydrodynamics and Ocean Engineering. Ocean Eng. http://dx.doi.org/10.1115/OMAE2022-78433, V05BT06A026.
Krata, P., Szlapczynska, J., 2018. Ship weather routing optimization with dynamic constraints based on reliable synchronous roll prediction. Ocean Eng. 150, 124–137.
Kuhlemann, S., Tierney, K., 2020. A genetic algorithm for finding realistic sea routes considering the weather. J. Heuristics 26, 801–825.
Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F., Ravuri, S., Ewalds, T., Eaton-Rosen, Z., Hu, W., Merose, A., Hoyer, S., Holland, G., Vinyals, O., Stott, J., Pritzel, A., Mohamed, S., Battaglia, P., 2023. GraphCast: Learning skillful medium-range global weather forecasting. arXiv:2212.12794.
Lang, X., Mao, W., 2020. A semi-empirical model for ship speed loss prediction at head sea and its validation by full-scale measurements. Ocean Eng. 209, 107494.
Lang, X., Wu, D., Mao, W., 2022. Comparison of supervised machine learning methods to predict ship propulsion power at sea. Ocean Eng. 245, 110387.
Liu, L., Shibasaki, R., Zhang, Y., Kosuge, N., Zhang, M., Hu, Y., 2023. Data-driven framework for extracting global maritime shipping networks by machine learning. Ocean Eng. 269, 113494. http://dx.doi.org/10.1016/j.oceaneng.2022.113494.
Madec, G., Bourdallé-Badie, R., Bouttier, P.-A., Bricaud, C., Bruciaferri, D., Calvert, D., Chanut, J., Clementi, E., Coward, A., Delrosso, D., Ethé, C., Flavoni, S., Graham, T., Harle, J., Iovino, D., Lea, D., Lévy, C., Lovato, T., Martin, N., Vancoppenolle, M., 2017. NEMO ocean engine. Notes du Pôle de modélisation de l'Institut Pierre-Simon Laplace (IPSL) v3.6-patch, http://dx.doi.org/10.5281/zenodo.3248739.
Mannarini, G., Carelli, L., 2019. VISIR-1. b: Ocean surface gravity waves and currents for energy-efficient navigation. Geosci. Model Dev. 12, 3449–3480.
Mannarini, G., Pinardi, N., Coppini, G., Oddo, P., Iafrati, A., 2016. VISIR-I: small vessels–least-time nautical routes using wave forecasts. Geosci. Model Dev. 9, 1597–1625.
Mannarini, G., Salinas, M.L., Carelli, L., Petacco, N., Orović, J., 2024. VISIR-2: ship weather routing in Python. Geosci. Model Dev. 17, 4355–4382.
McWilliams, J.C., 2000. Chapter 14 - formulation of oceanic general circulation models. In: Randall, D.A. (Ed.), General Circulation Model Development. In: International Geophysics, vol. 70, Academic Press, pp. 421–456. http://dx.doi.org/10.1016/S0074-6142(00)80062-5.
Mittendorf, M., Nielsen, U.D., Bingham, H.B., 2022. Data-driven prediction of added-wave resistance on ships in oblique waves—A comparison between tree-based ensemble methods and artificial neural networks. Appl. Ocean Res. 118, 102964. http://dx.doi.org/10.1016/j.apor.2021.102964.
Molland, A.F., Turnock, S.R., Hudson, D.A., 2011. Ship Resistance and Propulsion: Practical Estimation of Ship Propulsive Power. Cambridge University Press, New York.
Morales-Márquez, V., Orfila, A., Simarro, G., Marcos, M., 2020. Extreme waves and climatic patterns of variability in the eastern North Atlantic and Mediterranean basins. Ocean Sci. 16, 1385–1398. http://dx.doi.org/10.5194/os-16-1385-2020, URL: https://os.copernicus.org/articles/16/1385/2020/.
National Oceanic and Atmospheric Administration (NOAA), Beaufort wind scale. https://www.weather.gov/mfl/beaufort. (Accessed 10 July 2024).
National Oceanic and Atmospheric Administration (NOAA), Estimating wave height using wind speed during a tropical cyclone. https://www.vos.noaa.gov/MWL/201512/waveheight.shtml. (Accessed 10 July 2024).
National Oceanic and Atmospheric Administration (NOAA), 2024c. Gulf stream speed. https://oceanservice.noaa.gov/facts/gulfstreamspeed.html. (Accessed 10 July 2024).
NEMO TOP Working Group, 2022. TOP – Tracers in ocean paradigm – The NEMO tracers engine. http://dx.doi.org/10.5281/zenodo.1471700.
Perera, L.P., Soares, C.G., 2017. Weather routing and safe ship handling in the future of shipping. Ocean Eng. 130, 684–695. http://dx.doi.org/10.1016/j.oceaneng.2016.09.007.
Polvani, L.M., Scott, R., Thomas, S., 2004. Numerically converged solutions of the global primitive equations for testing the dynamical core of atmospheric GCMs. Mon. Weather Rev. 132, 2539–2552.
Precioso, D., 2023. Applications of machine learning and data science to the blue economy: sustainable fishing and weather routing (Ph.D. thesis). Universidad de Cádiz.
Precioso, D., Milson, R., Bu, L., Menchions, Y., Gómez-Ullate, D., 2024. Hybrid search method for Zermelo's navigation problem. Comput. Appl. Math. 43, 250.
Rasp, S., Hoyer, S., Merose, A., Langmore, I., Battaglia, P., Russel, T., Sanchez-Gonzalez, A., Yang, V., Carver, R., Agrawal, S., Chantry, M., Bouallegue, Z.B., Dueben, P., Bromberg, C., Sisk, J., Barrington, L., Bell, A., Sha, F., 2024. WeatherBench 2: A benchmark for the next generation of data-driven global weather models. arXiv:2308.15560.
Rinauro, B., Begovic, E., Mauro, F., Rosano, G., 2024. Regression analysis for container ships in the early design stage. Ocean Eng. 292, 116499.
Szlapczynski, R., Szlapczynska, J., Vettor, R., 2023. Ship weather routing featuring w-MOEA/D and uncertainty handling. Appl. Soft Comput. 138, 110142.
Techy, L., 2011. Optimal navigation in planar time-varying flow: Zermelo's problem revisited. Intell. Serv. Robotics 4, 271–283.
Townsin, R., Kwon, Y., 1983. Approximate formulae for the speed loss due to added resistance in wind and waves. Eng. Environ. Sci.
Tsai, C.-L., Su, D.-T., Wong, C.-P., 2021. An empirical study of the performance of weather routing service in the North Pacific ocean. Marit. Bus. Rev. 6, 280–292.
Uber Technologies, Inc., 2018. H3 - hexagonal hierarchical geospatial indexing system. Software URL: https://h3geo.org/docs/core-library/restable/.
Unidata Program Center, 2023. Network Common Data Form (NetCDF) User's Guide. University Corporation for Atmospheric Research (UCAR), Boulder, Colorado, URL: https://docs.unidata.ucar.edu/netcdf-c/current/.
Vancoppenolle, M., Rousset, C., Blockley, E., Aksenov, Y., Feltham, D., Fichefet, T., Garric, G., Guémas, V., Iovino, D., Keeley, S., Madec, G., Massonnet, F., Ridley, J., Schroeder, D., Tietsche, S., 2023. SI3, the NEMO Sea Ice Engine. http://dx.doi.org/10.5281/zenodo.7534900.
Vettor, R., Szlapczynska, J., Szlapczynski, R., Tycholiz, W., Soares, C.G., 2020. Towards improving optimised ship weather routing. Pol. Marit. Res. 27, 60–69.
Wang, Q., Danilov, S., Sidorenko, D., Timmermann, R., Wekerle, C., Wang, X., Jung, T., Schröter, J., 2014. The Finite Element Sea Ice-Ocean Model (FESOM) v.1.4: formulation of an ocean general circulation model. Geosci. Model Dev. 7, 663–693. http://dx.doi.org/10.5194/gmd-7-663-2014, URL: https://gmd.copernicus.org/articles/7/663/2014/.
Wang, H., Lang, X., Mao, W., 2021. Voyage optimization combining genetic algorithm and dynamic programming for fuel/emissions reduction. Transp. Res. D 90, 102670.
Wang, H., Lang, X., Mao, W., Zhang, D., Storhaug, G., 2020. Effectiveness of 2D optimization algorithms considering voluntary speed reduction under uncertain metocean conditions. Ocean Eng. 200, 107063.
Yang, Y., Zhang, Z., Zhao, J., Zhang, B., Zhang, L., Hu, Q., Sun, J., 2024. Research on ship resistance prediction using machine learning with different samples. J. Mar. Sci. Eng. 12, 556. http://dx.doi.org/10.3390/jmse12040556.
Young-Joong, K., 2005. A research on the approximate formulae for the speed loss at sea. J. Ocean Eng. Technol. 19, 90–93.
Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., Choi, Y., 2019. HellaSwag: Can a machine really finish your sentence? arXiv arXiv:1905.07830.
Zhao, L., Ohshima, T., Nagamochi, H., 2008. A* algorithm for the time-dependent shortest path problem. In: WAAC08: The 11th Japan-Korea Joint Workshop on Algorithms and Computation. p. 40.
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, 28–38.
Zis, T.P., Psaraftis, H.N., Ding, L., 2020. Ship weather routing: A taxonomy and survey. Ocean Eng. 213, 107697.
Zyczkowski, M., Szlapczynski, R., 2023. Collision risk-informed weather routing for sailboats. Reliab. Eng. Syst. Saf. 232, 109015.