WeatherRouting Bench 1.0: Towards Comparative Research in 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] and 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
Benchmark
Optimization
Github

ABSTRACT

This paper introduces `WeatherRouting Bench 1.0` to fill a gap in weather routing research: the lack of a common standard for comparing algorithms. We provide a website with code in Python that will assist researchers in using weather data, testing routes, and employing a fair cost function for comparing different methods: `http://benchmark.weathernavigation.com/`. We also suggest optimization problems based on real shipping routes, each including a standard route as a baseline for comparison. Furthermore, our paper investigates how the choice of a problem affects the optimization by analyzing how the average cost of a journey varies across different sailing speeds, regions of the ocean and times of the year. By offering this benchmark, we aim to make it easier for researchers to compare their work in weather routing. As a first entry to the benchmark, we propose A*-FMS: an algorithm based on A* graph search followed by a discrete Newton-Jacobi method, which combines the power of graph methods for global search and gradient descent methods for local optimization. The hyperparameters of this new method are discussed. With the best configuration and compared to the orthodromic route, the method reduces travel time by 1.4% when sailing at 12 knots, and 3.6% at 6 knots.

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 [30, 50]. The excellent review of Zis, Psaraftis and Ding [68] provides an extensive picture of the state of the art in weather routing up to 2020.

We can group weather routing algorithms into three categories. First, **evolutionary algorithms** [62, 34, 67, 24] represent the most exploratory approach, effectively identifying feasible routes in complex scenarios. However, they tend to be non-deterministic and lack guaranteed optimality [52]. Second, **graph search** techniques [22, 25, 69, 38] are known for their speed and exploitative nature, being the preferred method for obstacle avoidance [68]. Finally, **variational methods** [18, 53] excel in finding locally optimal paths, but their effectiveness is limited due to their dependence on a feasible initial solution [52]. Table 1 presents some of the most recent studies in weather routing, detailing the algorithms used and the journeys examined.

Assessing the effectiveness of a weather routing algorithm requires deciding on which journeys to test, and it is essential to understand two key concepts. First, an **Origin-Destination Pair (ODP)** identifies the start and end points of a journey, which, in practical applications, are maritime ports. In addition to the ODP, one must specify a departure date, and possibly a time. This in turn affects the weather and oceanographic conditions that will be found along the journey. In weather routing, a **route** is a solution for an ODP and specific departure date, specifying a path to follow based on factors such as safety, efficiency, and optimal utilization of oceanography and weather conditions.

A closer analysis of Table 1 reveals a significant challenge in weather routing research: each study uses a unique set of ODPs, complicating direct comparisons between algorithms. When studies do share an ODP, they often differ in departure dates, leading to distinct optimization scenarios. Additionally, the optimization objectives themselves vary across studies; some focus on minimizing travel time [53], while others aim to reduce fuel consumption [38]. Even when the optimization goals align, the methods used to calculate these costs differ, further complicating the evaluation of algorithm performance.

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. [38]. When assessing the efficiency of a new methodology, researchers can only score it against basic solutions, such as the orthodromic route [53, 7, 24, 67]. In some rare cases, they have access

[*]Corresponding author
✉ `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)
ORCID(s): `0000-0001-8919-0653` (J. Jiménez de la Jara); `0000-0003-3836-1429` (D. Precioso); `0009-0007-1333-4973` (L. Bu); `0000-0003-1457-9263` (M.V. Redondo-Neble); `0000-0001-5831-2056` (R. Ballester-Ripoll); `0000-0002-6890-6584` (D. Gómez-Ullate)

Table 1
Examples of recent weather routing studies, indicating the algorithms they used and the Origin-Destination Pairs (ODPs) they tested on.

Reference	Regions and ODPs	Algorithm
Mannarini et al. [38]	Mediterranean Sea (Porto Torres to Toulon, Monemvasia to Marmaris)	Graph Search VISIR-2
Precioso et al. [53]	North Atlantic Ocean (Charleston to Azores), Indian Ocean (Somalia to Myanmar), Caribbean Sea (Cancun to Charleston, Panama to Houston)	Hybrid Search
Charalambopoulos et al. [7]	Mediterranean Sea (Barcelona to Limassol, Barcelona to Thessaloniki, Barcelona to Alexandria)	Probabilistic roadmaps
Grandcolas [24]	North Atlantic Ocean (Cap Lizard to New York)	Evolutionary algorithm
Grifoll et al. [25]	North Atlantic Ocean (Boston to Plymouth), Mediterranean Sea (Tunis to Nice, Palma de Mallorca to Barcelona), others	A* graph search
Zhao et al. [67]	Indian Ocean (Singapore to Cape Town), Pacific Ocean (Shanghai to Los Angeles)	Particle swarm
Kuhlemann and Tierney [34]	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. [58]	Pacific Ocean (Taipei to Los Angeles, Tacoma to Kaohsiung)	*Non disclosed*
Gkerekos and Lazakis [22]	North Atlantic Ocean and Mediterranean Sea (Gulf of Guinea to Marseille)	A* graph search
Vettor et al. [62]	Atlantic Ocean (Portugal to Azores)	Evolutionary algorithm

to recorded data from a vessel journey [62, 22, 58]. Even when some algorithms are validated against real data, there is still a lack of comparison between different studies. This observation emphasizes the need for establishing standardized problems in weather routing research to facilitate more direct comparisons and benchmarking of algorithms. Such standardization could accelerate progress in the field by enabling researchers to build upon a common foundation of scenarios and data sets.

One notable example of the importance of standardized benchmarks can be seen in the field of Natural Language Processing (NLP) with the development of benchmarks for Large Language Models (LLMs). Benchmarks such as the MMLU [27] and HellaSwag [66] have been essential in the evaluation of LLMs like GPT-3 [5], Mixtral [31], and LLaMA [56]. It has been one of the main components that allowed the popularity and growth of this field and will help improve many aspects, such as reasoning, understanding, and text generation.

Another recent example closer to weather routing is Google's WeatherBench II [54]. It has set a standard for evaluating weather prediction models by offering a common dataset and evaluation metrics. This standardization has advanced weather forecasting by enabling direct comparisons and highlighting the strengths and weaknesses of various models, such as IFS HRES [14], ERA5 Forecast [28], and Graphcast [35] - a new method based on machine learning that outperforms traditional models in some aspects. This comparative system also encourages competition, which drives a considerable improvement in the field.

This paper's primary contribution is the development of benchmarking standards. We introduce **`WeatherRouting Bench 1.0`**. Every instance in our benchmark specifies origin and destination ports, a starting time, and includes data on ocean currents and waves from the Copernicus Maritime Service [12, 13]. The necessary Python code for data acquisition is made available in a public website[1]. Additionally, researchers can submit their proposed optimal routes for each problem instance inside our benchmark. Submissions undergo automated scoring and ranking, offering a platform for comparative analysis against leading methodologies in a standardized setting. Initially, the evaluation will focus on reducing travel times, taking into account only the effect of waves and currents, with plans to later incorporate wind effects and fuel consumption metrics.

Another critical aspect of our contribution addresses the lack of diversity in weather conditions. Traditional studies often limit their focus to a single departure date per ODP, overlooking the inherent variability of weather conditions that require therefore adaptable and robust algorithms. `WeatherRouting Bench 1.0` incorporates multiple departure dates for each ODP. This allows for the evaluation of algorithms across 52 different weeks for the same ODP (every week in a full year, thus enabling seasonal studies), providing comprehensive insights into their performance and showcasing their full potential.

Finally, serving as the first use-case of `WeatherRouting Bench 1.0`, this paper introduces a new version of one of the most popular algorithms for weather routing: the A* graph search [38, 25, 22, 68]. 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. In addition, a refinement step is added after A*, based on the Ferraro-Martín de Diego-Sato algorithm (FMS)

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

[18, 17], which modifies the solution given by the A* using calculus of variations, moving its waypoints to a local optimum while visually smoothing the whole route. The FMS is an exploitative algorithm that has been recently applied in conjunction with a shooting method for weather routing [53] and showed great potential. This paper shows that A* provides a good initial solution for the FMS, but the second step in the optimization pipeline greatly improves the results since trajectories are not constrained anymore to run on a discrete mesh.

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. This is followed by an explanation on how we download and process weather data in Section 3. We then introduce `WeatherRouting Bench 1.0` in Section 4 as a collection of instances with variation on different elements that define each instance. We describe all necessary elements required for a full specification of a problem instance: origin and destination, vessel's characteristics, consumption models, etc. Section 5 explains the A*-FMS optimization algorithm that serves as a first entry for `WeatherRouting Bench 1.0`. In Section 6 we apply the A*-FMS optimization algorithm to the whole `WeatherRouting Bench 1.0` dataset. We will study how the optimization results are influenced by the characteristics of every optimization problem such as the ODP, vessel speed, the relative impact of currents and waves, and the seasons.

2. Optimization problem

An *instance* of a weather routing problem is defined by the following components:

- A pair of departure and destination ports (ODP),
- a specified departure date and time,
- vessel characteristics,
- a prescribed cruising speed,
- relevant meteorological information (e.g., waves, currents),
- a cost function that depends on the aforementioned parameters,
- decision variables available to the algorithm to minimize the cost function.

Weather routing algorithms are each designed for a specific kind of optimization problem. A robust optimization algorithm should be capable of handling all optimization problems of a same type, provided the set of available decisions remains unchanged. For `WeatherRouting Bench 1.0`, we assume that vessels sail at a constant speed over water v_{wtr}, implying that the vessel's engine power does not vary along the route.

A solution of the optimization problem will be given as a polygonal curve defined by a sequence of waypoints $\{q_i\}$, where $q_i = (x_i, y_i, t_i)$, $i \in [N]$, and N is the total number of waypoints along a route. Here, q_1 is the starting location, and q_N is the destination. Each waypoint is a triple, where each entry represents the longitude, latitude, and timestamp at the i-th waypoint, respectively. We will adhere to the convention of specifying the route by providing the position of the vessel at equally fixed time intervals $\Delta t = \{15, 30, 60\}$ min for the three cases $v_{\text{wtr}} = \{24, 12, 6\}$ knots. The decision variables for the optimization problem are thus to determine the course heading at every Δt time interval.

Although vessels travel at a prescribed speed v_{wtr} over (calm) water, their actual speed with respect to ground v_{grd} is affected by ocean waves and currents. The total travel time will depend thus on these navigation conditions and the chosen route, and it will be the **cost function** we aim to optimize in this first version of `WeatherRouting Bench 1.0`. To calculate this time, we need to understand how environmental factors affect vessel speed, especially speed loss due to wave resistance and changes caused by ocean currents.

A more realistic optimization problem would be to minimize fuel consumption subject to a prescribed maximum travel duration, allowing both changes in course and engine power along the route. Future versions of our weather routing benchmark will incorporate more realistic optimization problems and cost functions such as the one mentioned above, and possibly extending it also to safety parameters. For this first version, we believe the proposed optimization problem is already rich enough to constitute a valid starting point.

2.1. Effect of waves on vessel speed

First, we will analyze the impact of wave-induced speed reduction. Since the 1970s, both academia and industry have put a lot of effort into studying this effect. Various methods have been developed, ranging from statistical and regression models based on wave tank experiments [29, 16, 65, 57] to advanced fluid dynamics and potential flow models [32]. Recently, data-driven methods, including Machine Learning and Neural Networks [40, 64, 33], have further improved our ability to predict wave resistance and the resulting speed losses.

We have chosen the Townsin-Kwon [57] model in Molland et al. [41], as it has valid results for a relatively small set of environmental parameters: significant wave height h_{wav} (in meters) and wave incidence angle θ (in degrees). It also depends on the speed over water v_{wtr} (in meters per second), its length L (in meters), displacement ∇ (in cubic meters) and block coefficient c_B.

Young-Joong [65] later updated the Townsin-Kwon equations of percentage speed loss over water:

$$c_w := \frac{\Delta v_{\text{wtr}}}{v_{\text{wtr}}} \cdot 100\% = c_\beta c_u \alpha \tag{1}$$

where c_β is the weather reduction coefficient, c_u is the speed reduction coefficient, α is the correction factor, and we denote the percentage speed loss by c_w. These dimensionless coefficients are calculated and adapted from Molland et al. [41] as explained below.

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

$$2c_\beta := \begin{cases} 2 & 0^\circ \leq \theta < 30^\circ \\ 1.7 - 0.03 \cdot (\mathrm{BN} - 4)^2 & 30^\circ \leq \theta < 60^\circ \\ 0.9 - 0.06 \cdot (\mathrm{BN} - 6)^2 & 60^\circ \leq \theta < 150^\circ \\ 0.4 - 0.03 \cdot (\mathrm{BN} - 8)^2 & 150^\circ \leq \theta \leq 180^\circ \end{cases} \tag{2}$$

The Beaufort number is a dimensionless quantity usually derived from wind speed [43], but also related to h_{wav} by using the relation between wind speed and wave formation [44] (since the most important factor causing waves is surface wind):

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

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

$$c_u = 0.7\ \mathrm{BN} + \frac{\mathrm{BN}^{13/2}}{\theta \nabla^{2/3}}; \quad \theta = 22\ \mathrm{m}^{-2}. \tag{4}$$

Last, the correction factor α depends on the vessel's Froude Number Fr and its block coefficient c_B. For the purpose of our simulation, we take the formula derived from Molland et al. [41] for a block coefficient $c_B = 0.6$:

$$\alpha = 2.2 - 2.5\ \mathrm{Fr} - 9.7\ (\mathrm{Fr})^2, \tag{5}$$

In the above equation, Fr is a dimensionless quantity defined as:

$$\mathrm{Fr} = \frac{v_{\mathrm{wtr}}}{\sqrt{gL}}, \tag{6}$$

where v_{wtr} is the vessel speed over water, L is the length of the vessel and g is the gravity constant.

Note that Molland et al. [41] and Young-Joong [65] model wave effect as a percentage reduction in vessel's speed over water. The only effect considered in this paper is the so called *involuntary speed reduction*, which occurs by the pure mechanic effect of added resistance due to waves. In addition, the previous authors also consider a *voluntary speed reduction*, which results from the captain's decision to lower the thrust to reduce vessel's speed in the presence of high waves, both for safety and efficiency reasons.

A requirement to the optimization algorithm we present in this paper (Section 5.2) is to evaluate first- and second-order derivatives of the cost function with respect to the decision variables. The coefficient c_β in equation (2) is a piece-wise constant function of wave incidence angle θ, and this lack of continuity and smoothness causes numerical instability of the optimization algorithm. To prevent this, we have substituted c_β in equation (2) by a smooth function, as seen in Figure 1. The smooth $\hat{c}_\beta$ is given by

$$2\hat{c}_\beta = c - b \cdot (\mathrm{BN} - a)^2, \tag{7}$$

where

$$a = 6 \sin^{(2/3)}(\alpha/2) + 2, \tag{8}$$

$$b = \frac{1}{40}\big(1 + \sin(1.2\alpha) - \cos(1.2\alpha)\big), \tag{9}$$

$$c = 2 - 1.6 \sin(\alpha/2). \tag{10}$$

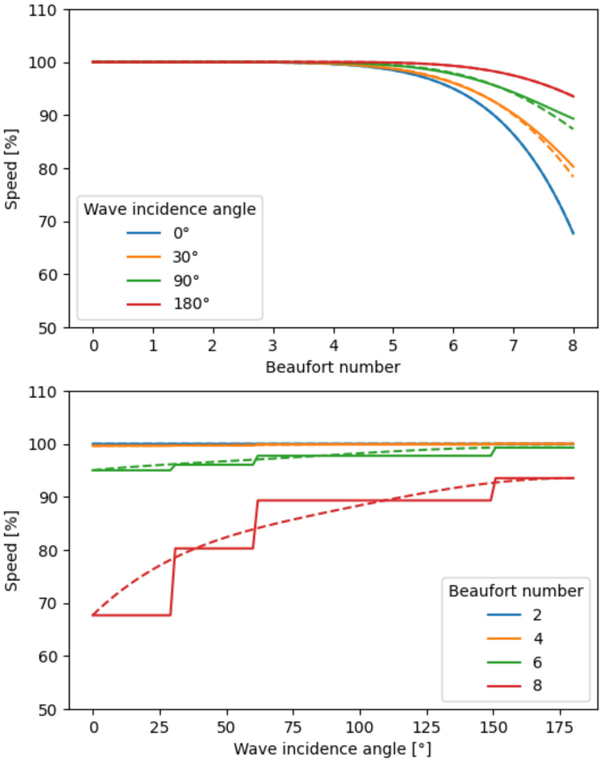


Figure 1: Comparison of $\hat{c}_\beta$ (dotted line) defined in (7) versus the non-smooth c_β from [41] (solid line).

2.2. Effect of ocean currents on vessel speed

The other major component affecting vessel speed is ocean currents. Unlike waves, which are always detrimental, currents can either increase or decrease the vessel's ground speed $\vec{v}_{\mathrm{grd}}$.

To model the effect of currents on the vessel's speed, in line with other studies we simply consider vector addition:

$$\vec{v}_{\mathrm{grd}} = \vec{v}_{\mathrm{vessel}} + \vec{v}_{\mathrm{cur}}.$$

Let $v_{\mathrm{cur},\parallel}$ and $v_{\mathrm{cur},\perp}$ denote the components of the current velocity that are parallel and perpendicular to the vessel's direction, respectively. We have:

$$v_{\mathrm{grd}} = v_{\mathrm{cur},\parallel} + \sqrt{\left(c_w \cdot v_{\mathrm{wtr}}\right)^2 - v_{\mathrm{cur},\perp}^2} \tag{11}$$

The combined effect of waves (1) and currents (11) determine the actual speed v_{grd} at a given position and time. Let $v_{\mathrm{grd},i}$ be the speed of the vessel at position q_i and time t_i. We are assuming that a value of the waves height and direction and the ocean currents are available at every position q_i in the ocean. Weather and ocean data as provided by forecasting services (see Section 3) are given on a grid as a NetCDF file, so interpolation functions are necessary to retrieve weather data at an arbitrary position. We discuss the interpolation methods employed in Appendix A.

We assume that the weather conditions between q_i and q_{i+1} remain constant, and thus the polygonal curve that defines the route is traversed in a piecewise linear way. For a more robust numerical scheme, we consider that the speed of the vessel in covering the segment from q_i to q_{i+1} is given by the average between $v_{\mathrm{grd},i}$ and $v_{\mathrm{grd},i+1}$, and thus all waypoints in the route should satisfy the condition

$$\Delta d_i = \frac{1}{2}\left(v_{\mathrm{grd},i} + v_{\mathrm{grd},i+1}\right)\,\Delta t, \quad i = 1, \dots, N-1, \quad (12)$$

where Δd_i is the geodesic distance between two consecutive waypoints q_i and q_{i+1} (for simplicity, we calculate it using the haversine formula). A route with waypoints $\left\{q_i = (x_i, y_i, t_i)\right\}_{i=1}^{N}$ must satify (12) to be compatible with the vessel cruise speed v_{wtr} and the weather conditions along the route. In our website we provide functions to test the feasibility of every possible route. We also provide functions to reparametrize polygonal trajectories so that the reparametrized waypoints are equally spaced in time while preserving the feasibility of the route.

This completes the definition of the optimization problem to be tackled in `WeatherRouting Bench 1.0`. Future versions of the benchmark will include other cost functions, such as fuel consumption, GHG emissions or safety at sea, and other decision variables (freedom to change both engine power and direction).

3. 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. This Section introduces the real data used in `WeatherRouting Bench 1.0`, which uses NEMO 3.2 model to obtain currents data and MFWAM 3.3 for waves.

3.1. 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 [39]. By mathematically modelling 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 [3] 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 modelling, 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 [63].
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 [51].

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.

3.2. Nucleus for European Modelling of the Ocean (NEMO)

The Nucleus for European Modelling of the Ocean [37] represents an important component for the global oceanographic research. Developed through a collaborative effort by several leading European research institutions, it is a versatile modelling 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 [3].

NEMO's framework offers a holistic approach to ocean modelling by integrating multiple key components. It simulates oceanic physical processes using advanced numerical methods to understand currents, temperature, salinity, and sea level changes [37]. Additionally, NEMO handles the complex interactions between the ocean and sea ice, including formation and melting, and their effects on circulation [61]. It also explores marine ecosystems and bio-chemical cycles to study the ocean's contribution to the carbon cycle and environmental responses [46].

3.3. Météo-France Wave Model (MFWAM)

The Météo-France Wave Model (MFWAM) is an advanced, flexible modelling 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 [15] computing code, incorporating dissipation terms developed by Ardhuin et al. [4]. 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" [49, 13].

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.

3.4. Data processing

The maintenance of waves (MFWAM) and currents (NEMO) data are undertaken by Copernicus [8], 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 [12]. Within this project, the focus is primarily on the current data, which is analyzed in terms of its vertical (v_0) and horizontal (u_0) 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 [12]. However, only the surface level is utilized for this analysis. Considering that the usual draught of a container vessel is 12m [55], the currents vary 0.01 m/s on average between the surface layer and at 12m depth, which has a negligible effect compared with a vessel's typical speed.

`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 [13]. 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 hours.

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 hours. 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 [60]. 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.

3.5. Interpolation

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 5.2. 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 A.

4. Weather Routing Bench 1.0

Having introduced what constitutes a weather routing problem, we can now define the set of instances that make the first version of `WeatherRouting Bench 1.0`. 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 [36]. Maritime networks reveal various aspects of global shipping, including mapping traffic densities to show the spatial patterns of vessel movements [10]. 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 [1, 2, 21].

Table 2
Origin-Destination Pairs in `WeatherRouting Bench 1.0`: regions they connect and orthodromic distances. Each ODP yields two possible routes: from port 1 to port 2, and vice versa.

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

Table 3
Parameters of every instance of `WeatherRouting Bench 1.0`.

Journey	
ODP	See Table 2 (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 over Water	$v_{\text{wtr}} = \{6, 12, 24\}$ kn (constant)
Optimization Problem	
Cost Function	Travel time affected by waves and currents.
Solution	Polygonal curve with waypoints (latitude, longitude) specifying vessel position at constant time intervals.

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 2. 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, in this first version of the benchmark we will use a typical container ship. Its parameters are listed in table 3. We assume the vessel can achieve constant power delivery for the entire duration of the journey, and as such, the vessel sails at constant speeds over water, v_{wtr}. We considered three different speeds, 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 specifying the vessel's position (latitude, longitude) at constant time intervals of $\Delta t = \{15, 30, 60\}$ min when the vessel has speed $v_{\text{wtr}} = \{24, 12, 6\}$ kn, respectively. With this choice, the average distance between waypoints is around 10 km, which is of the same order as the grid spacing of weather data. 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.

5. Optimization Algorithm

5.1. A* graph search

Graph optimization is a powerful mathematical technique widely used in the field of weather routing [68, 25]. 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 5.2.

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 sensible graph would be to consider the data grid provided in our meteorological data; the distance between points in currents data being $1/12° \simeq 0.083°$ (approximately 10 km at the equator). This results in a grid of size 4320 ×

Table 4
H3 resolutions. Data from Uber Technologies, Inc. [59].

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

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, (aka, Mercator Projection) has limitations due to the Earth's spherical shape, which requires significant distortion to fit the grid.

To address this issue, we use instead an hexagonal grid provided by the H3 library developed by Uber Technologies [59]. As shown in Figure 2, hexagons accurately cover the round shape of the Earth, known as a hexasphere. 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 4.

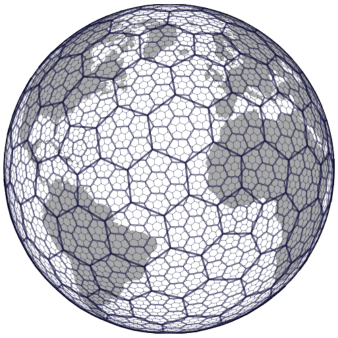


Figure 2: Partition of the Earth into hexagonal grids of different sizes, using H3. Image from Uber Technologies, Inc. [59].

Edges of the graph

Each hexagon of the H3 grid has six neighbors, allowing for two more possible directions per node compared to a traditional square grid. This is important for real applications of weather routing, as vessels take smooth turns. A square forces 90° turns while an hexagon 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 ***N*-order neighbors**, effectively providing $6 \cdot \left(1 + \sum_{k=0}^{N-1} k\right)$ possible directions and reducing the angle of course changes, as depicted in Table 5.

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 Figure 3 for first and second order neighbors.

Table 5
Graph characteristics dependence on the N-order neighbors.

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

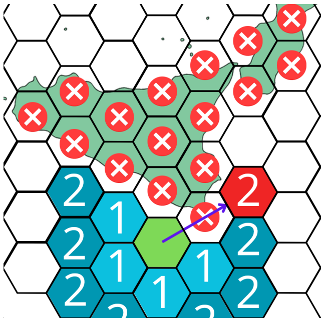


Figure 3: The representation demonstrates how the algorithm can jump over several neighboring nodes, enhancing the flexibility of the route. It also shows the two techniques to ensure land avoidance: (1) Hexagons located directly on land are removed, and (2) edges that cross land are also removed.

Heuristic

With the graph constructed, the goal of our weather routing algorithm is to find the minimal path between two nodes ($n_{\text{start}}, n_{\text{end}}$). 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 [9]. 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 [26], 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:

$$f(n) = g(n) + h(n), \tag{13}$$

where $g(n)$ is the cost of the path from n_{start} to n, and $h(n)$ is the heuristic cost of the path from n to n_{end}. In our particular case, we choose

$$h(n) = \frac{d(n, n_{\text{end}})}{\bar{v}_{\text{short}}}, \tag{14}$$

where $d(n, n_{\text{end}})$ is the haversine distance between nodes n and n_{end} and $\bar{v}_{\text{short}}$ is the average ground speed of the vessel computed along the route of shortest distance (orthodromic).

A⋆ takes into account both the explored path and an approximation of the remaining path. Thus, A⋆ provides an optimal solution as long as the heuristic is admissible, i.e. it does not overestimate the real cost from n to n_{end}. Although A⋆ is guaranteed to find an optimal route if the heuristic is admissible, the quality of the heuristic affects the number of nodes explored by the algorithm, so the complexity of A⋆ depends on the heuristic chosen. In the worst-case scenario, a poor heuristic will force A⋆ to explore every node in the graph, similarly to Dijkstra's algorithm.

There are several variations of A⋆ that can help the algorithm to converge faster. One of them is weighted A⋆ [11], where the cost function is defined as:

$$f(n) = g(n) + w \cdot h(n). \tag{15}$$

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 n_{start} to n to be disregarded, resulting in a Greedy Best First Search [19]. Higher values of w are expected to speed up the algorithm, but no longer guarantee an optimal result, even when using admissible heuristics. Can we argue that our heuristic is approximately admissible? Otherwise this discussion is less relevant.

5.2. 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 5, 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, we employ the Ferraro-Martín de Diego-Sato (FMS) algorithm [18, 17]. This algorithm operates by numerically solving a Boundary Value Problem (BVP) associated with the Zermelo's Navigation Problem. FMS approach enhances the initial path produced by A⋆ by smoothing course corrections and improving overall route efficiency, finding local minimums of the cost function.

The FMS algorithm is a numerical method used to iteratively move a finite number of points along a route, denoted as $\{q_i\}_{i\in[n]}$, to a set of optimal points $\{q_i^*\}_{i\in[n]}$ that minimizes the cost within a neighborhood for each point q_i. This iterative process involves solving the following equation:

$$\begin{aligned} &D_2T(q_{i-1}, q_i) + D_1T(q_i, q_{i+1}) \\ &+ \left(D_{22}(q_{i-1}, q_i) + D_{11}T(q_i, q_{i+1})\right)(q_i^* - q_i) = 0 \end{aligned} \tag{16}$$

where D_k represents the partial derivatives with respect to the k-th argument of the cost function T, and $D_{k,l} = D_k \circ D_l$. The functional T can be derived from the travel time Δt in Equation (12), which is given by:

$$T(q_i, q_{i+1}) = \frac{2\Delta d_i}{v_{\text{grd},i} + v_{\text{grd},i+1}} \tag{17}$$

In this context, Δd_i represents the haversine distance between two consecutive waypoints q_i and q_{i+1}, while $v_{\text{grd},i}$ and $v_{\text{grd},i+1}$ denote the speeds with respect to ground at these points. For a comprehensive explanation of the underlying mathematical principles and detailed derivations, please refer to Ferraro et al. [18].

The FMS algorithm comprises three hyper-parameters: damping, maximum iterations, and early stopping, which collectively determine the movement of each point q_i in one iteration and the total number of iterations required to reach the local minima q_i^*. This can be best understood by drawing an analogy to Neural Networks: **damping** is akin to the *learning rate*, as it dictates the step size taken toward a local minimum in each iteration. This hyper-parameter provides a balance between the rate of convergence and the risk of overshooting [6].

Maximum iterations are a parallel to the training epochs in a neural network. Given that it is impossible to ensure convergence to a local minimum, the FMS algorithm could theoretically run indefinitely. Therefore, setting a predetermined limit on the number of iterations is crucial. Moreover, an **early stopping** mechanism is implemented to avoid unnecessary computations when the FMS algorithm is near the local minimum. This criterion halts the algorithm if the cost function does not improve over a specified number of consecutive iterations. These last two hyper-parameters effectively reduce computational time by terminating the algorithm once significant reductions of the cost function are unlikely. Further details can be found in Precioso et al. [53] and the references therein.

In Figure 4, we see an A⋆ solution with the sudden change in directions along the route. After applying few hundred iterations of the FMS algorithm, we can see that these sharp turns have been smoothed out. Not only does FMS provide a smoother trajectory, but is also guaranteed to converge to a locally optimal solution of the variational problem, as proved in Ferraro et al. [17]. The shortcoming of FMS 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. [53] 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 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 exploitation that is guaranteed to converge to a local minimum. We have named this new algorithm A⋆-FMS.

6. Results

After performing a hyper-parameter search for the first week of each instance, we selected the configuration of our

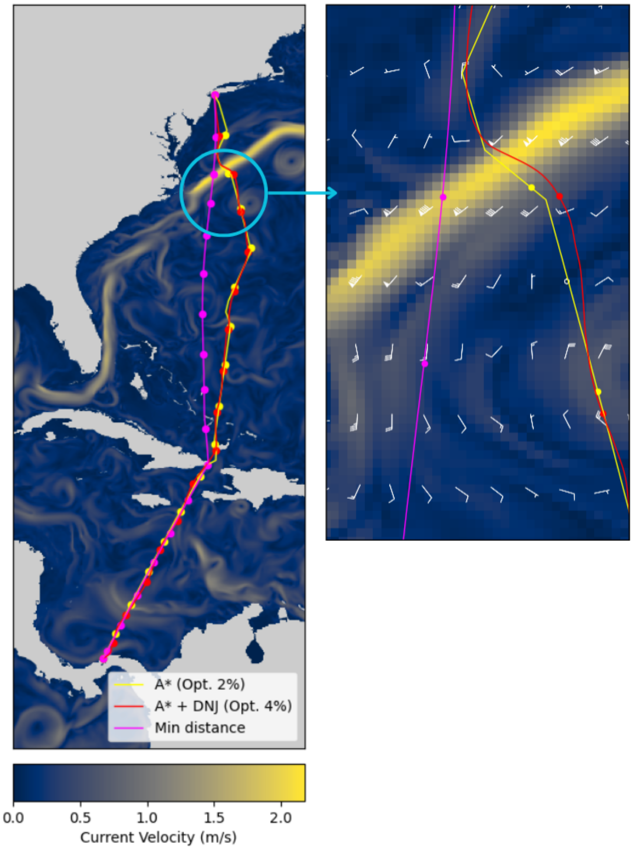


Figure 4: Route comparison between orthodromic route (purple), A⋆ (yellow) and A⋆-FMS (red). One can see that after applying FMS, the 'jagged' turns from A⋆ are smoothed out, as we move from the discrete space of A⋆ to a continuous space.

algorithm by considering the trade-offs between compute time and relative route savings. We relegate the discussion of the hyper-parameter tuning of A⋆-FMS to Appendix B. The A⋆ algorithm used for the remainder of this paper is configured with the following parameters: grid resolution of 4, 3rd-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 Section B.

There are five pairs of ports, introduced in Table 2, that can be travelled in both directions, totalling 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 of `WeatherRouting Bench 1.0`. 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.

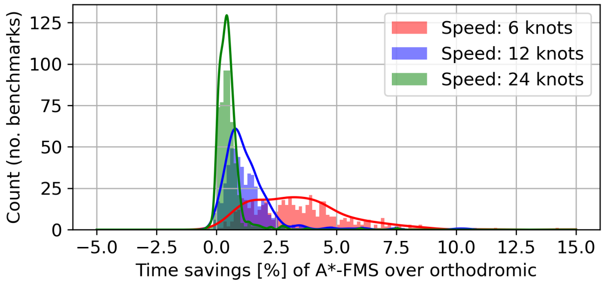


Figure 5: Time savings of our A⋆-FMS algorithm compared to the orthodromic route. Histogram and best-fit KDE univariate distribution for each sail speed.

6.1. Vessel speed

The first study conducted with these results examines how the gains of our A⋆-FMS algorithm are affected by vessel speed. Figure 5 shows a histogram of gains across all instances based on vessel speed over water. 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 A⋆-FMS solution does not reduce travel time w.r.t. 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 results with A⋆-FMS compared to following the orthodromic route. These losses are marginal, the lowest being -0.4% (20 minutes 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 A⋆-FMS algorithm. In contrast, at lower speeds, A⋆-FMS 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. They confirm the well known fact that the influence of weather increases at lower cruising speeds.

6.2. 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 Figure 6, we compare the overall gains (travel time reduction relative to the orthodromic route) of our A⋆-FMS algorithm for each set of ODPs, sorted by direction of travel, when sailing at 6 knots. The boxplots show the distribution of savings over 52 weeks for each ODP. The general trend of these boxplots is similar for vessel speeds of 12 and 24 knots, with gains inversely proportional to speed.

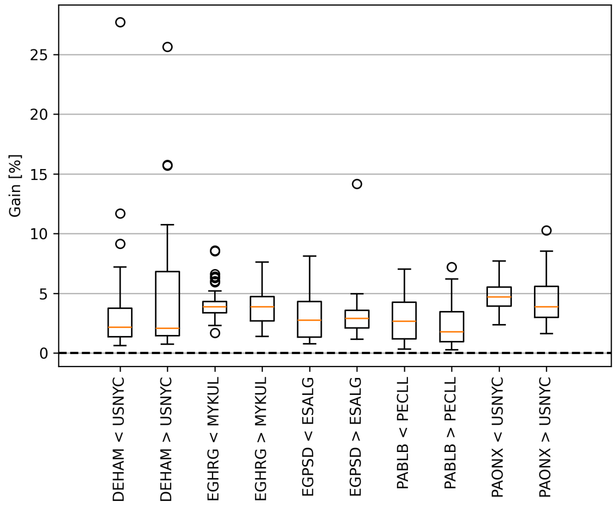


Figure 6: Percent reduction in travel time achieved by the A*-FMS algorithm compared to the orthodromic route at 6 knots. ODPs are sorted by decreasing distance along the X-axis.

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 specially in the route across the North Atlantic (DEHAM-USNYC). These are due to extreme weather events and will be analyzed in Section 6.3.

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 2 for distances). Indeed, the longest routes present the largest outliers, demonstrating that the algorithm can exploit certain favourable conditions when the weather is advantageous.

In Section 6.1, we noted that A*-FMS 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.

Table 6
Gains of A*-FMS 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)

6.3. 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.

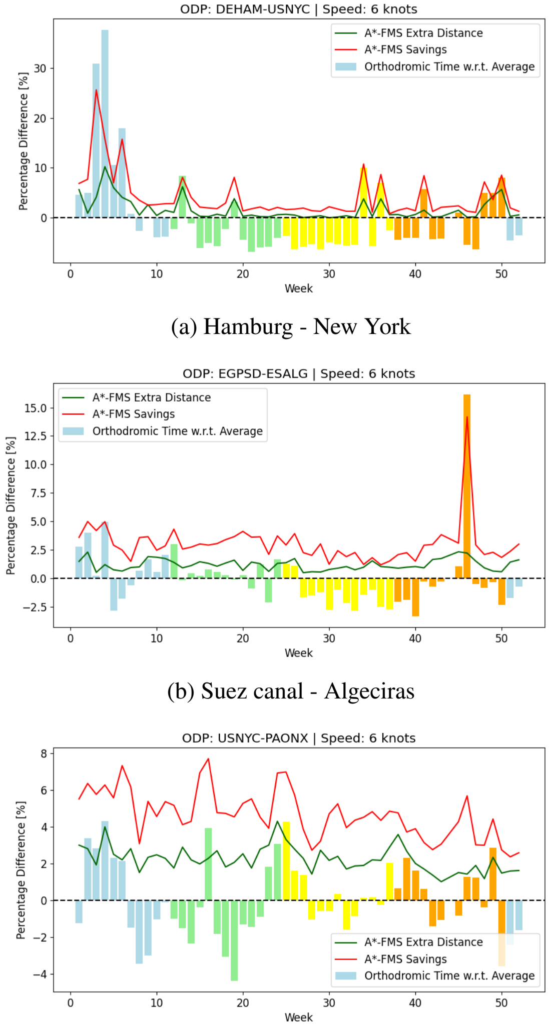
We identify greater savings across all speeds during winter, and less so in spring, compared to summer and autumn. In fact, gains in winter almost double the ones in other seasons. The standard deviation of winter is also the highest, implying that this is the season with the largest weekly differences,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

To better asses the impact of seasonality, we focus on how navigation conditions affect the fixed shortest distance route across the year. In Figure 7 the vertical bars show the relative variation of travel time for each week with respect to the year average. Solid lines show the gains of A*-FMS for every week and the extra distance covered.

Figure 7a represents Hamburg (DEHAM) to New York (USNYC), crossing the North Atlantic ocean. During winter (coloured 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 year average. With such a great discrepancy in time, A*-FMS 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 A*-FMS. 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 [42, 23].

Such extreme wave climates are also observed in the Mediterranean Sea. This fact is reflected in Figure 7b for Egypt (EGPSG) to Spain (ESALG), where an extreme event during summer time causes one orthodromic route to take 15% more time than usual, allowing A*-FMS 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 Figure 7c 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.



(a) Hamburg - New York

(b) Suez canal - Algeciras

(c) New York - Panama canal

Figure 7: Percentage difference between the orthodromic route per week to yearly average, and A*-FMS compared to orthodromic with respect to time and distance. Each week is coloured by season, starting with winter.

Weather routing algorithms such as A*-FMS 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 A*-FMS algorithm consistently finds routes that are on average 3% faster than the standard geodesic route.

6.4. 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.

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 [45]. Following these prior assumptions, we expect waves to have a larger impact, while currents become more relevant for slow steaming.

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

Speed	Gains avg. (std.) (%)		
(knots)	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 consider thus three separate speed reduction models that consider only waves, only currents or both. We compare the gains of our A*-FMS algorithm for all of these experiments in Table 7.

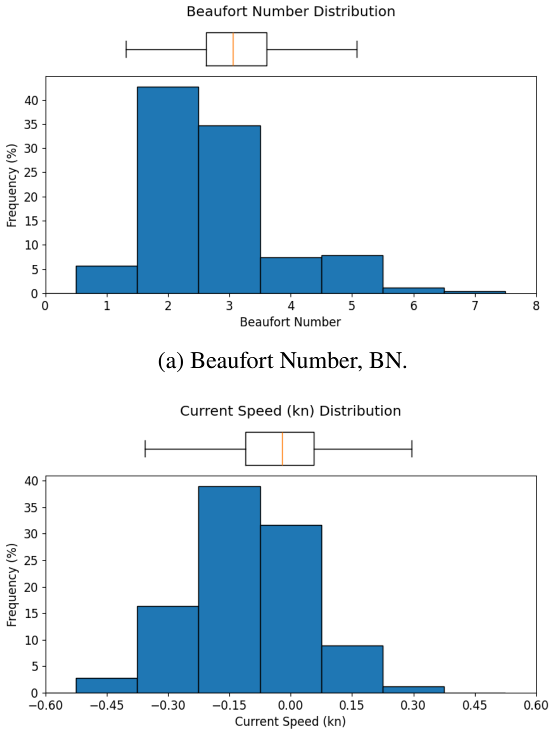
We found that for every vessel speed, the model that takes into account only waves achieves a smaller reduction than the one that considers only currents, while it is natural that both are dominated by the model that considers both effects. However, the standard deviation of the A*-FMS gains for the 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 fact that ocean currents have a larger variation per unit distance.

To study this effect in further detail, we consider orthodromic routes, and represent the distribution of both Beaufort numbers and speed of ocean currents. Beaufort is strongly related to wave height by equation (3), and we can see its effect on speed reduction in Figure 1, according to the model used in this study.

On Figure 8a, we see that 97% of the orthodromic routes' waypoints find a BN lower than five. The average speed reduction due to is waves minimal, around 0.3% - a percentage similar to the gains achieved by our A*-FMS algorithm. 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 shown in Figure 8b. 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 (wave 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.



(a) Beaufort Number, BN.

(b) Ocean current speed relative to the heading of the vessel. Positive values representing favourable currents.

Figure 8: Distribution of values encountered across every orthodromic route for all ten ODPs across the 52 weeks of the year 2023.

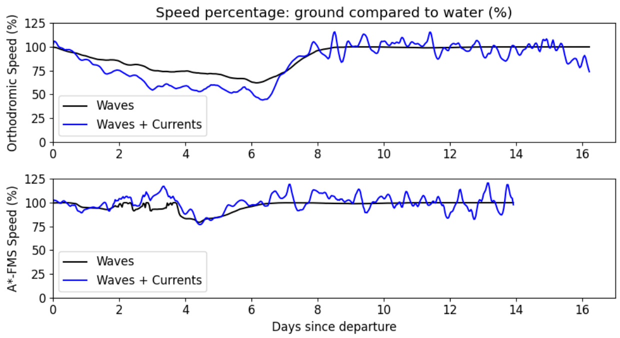


Figure 9: Percentage loss in sailing speed due to waves and ocean currents for the route from Suez Canal (EGPSD) to Spain (ESALG) during week 46 of 2023, at a sailing speed of 6 knots. The figure compares the shortest distance route (above) and the optimized route using A⋆-FMS (below).

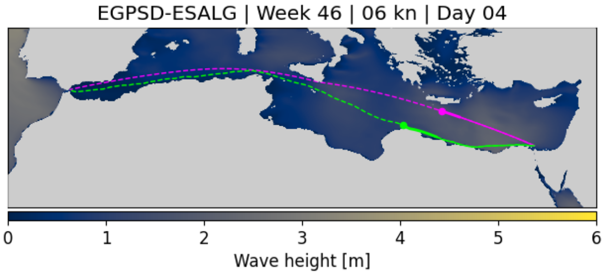


Figure 10: Route from Suez Canal (EGPSD) to Spain (ESALG) during week 46 of 2023, at a sailing speed of 6 knots. The figure compares the orthodromic route (magenta) and the optimized route using A⋆-FMS (green).

6.5. 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 Figure 7b. Figure 9 shows the percentage loss in sailing speed due to waves and ocean currents for both the orthodromic route and the one optimized using A⋆-FMS. Both routes face adverse weather conditions at the start of the journey. However, the A⋆-FMS solution takes a detour to follow favourable 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 to 50% of its original speed. This prolonged exposure to bad conditions force the vessel to spend a longer time in an unfavourable area, adding to the delay. As a result, the shortest distance route is covered in 390 hours for a total distance of 3533 km ($\bar{v}_{\mathrm{grd}} \approx 5$ kn), while the A⋆-FMS optimized route covers 3611 km in 335 hours ($\bar{v}_{\mathrm{grd}} \approx 6$ kn). Figure 10 shows the trajectory traversed by both routes. 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 A⋆-FMS 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 of `WeatherRouting Bench 1.0` will include more oceanographic and weather variables, as well as more complex and realistic models for the cost function.

7. Conclusion

This study introduced `WeatherRouting Bench 1.0`, a benchmarking platform for academics and industry partners to contribute to the weather routing problem. This platform offers easily accessible weather data files for download, available in our website[2]. It also includes auxiliary functions for interpolation of weather data, speed reduction models, scoring functions and reparametrization algorithms for other researchers willing to tackle weather routing optimization problems and score their proposed algorithms against the benchmark.

This first version of the benchmark is composed of five ODPs in different oceans connecting major hubs, as well as weather data covering a full year, thus allowing for

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

both space and time variability. Registered users will be able to submit their optimized routes and score them, thus contributing to the common knowledge advancement of the field. The optimization problem for this initial version of the benchmark involves minimizing travel time at a fixed speed over water. Actual vessel speed is affected by ocean waves and currents according to a simple model, for which we provide the relevant code and explanations.

A component to improve in `WeatherRouting Bench 1.0` are 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 [29, 16]. 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 [29]. 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 [47, 48]. Additionally, modeling ship motions can identify hazardous conditions that might endanger crew or cargo, especially for high-sensitivity cargo requiring specialized handling.

Another particular challenge faced by weather routing is quantifying uncertainty in weather forecasts. In `WeatherRouting Bench 1.0` 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 [8], with decreasing accuracy further ahead. Thus, uncertainties in weather forecasts demand robust algorithms and benchmarks that are able to quantify and handle the uncertainty which is inherent to weather prediction.

As the first entry to `WeatherRouting Bench 1.0`, this paper also introduces a new optimization method, A*-FMS. This is a combined algorithm, which 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.

As we have developed a platform to accelerate research and implementation of weather routing, we plan to address these shortcomings in `WeatherRouting Bench 2.0`. We will introduce more accurate ship dynamics and power output modeling, as well as improved cost functions that better reflect real-world costs and operational challenges. With these additions, we aim to provide an even more comprehensive set of optimization problems to better evaluate and score different weather routing algorithms.

CRediT authorship contribution statement

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

Acknowledgements

This publication is funded by TED2021-129455B-I00 project, funded by MICIU/AEI/10.13039/501100011033 and the European Union "NextGenerationEU/PRTR". JJ and VR would like to thank Agencia Estatal de Investigación for their support. 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 research of DP, RM, RB and DGU is supported by the BBVA Foundation via the project "Mathematical optimization for a more efficient, safer and decarbonized maritime transport". 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.

References

[1] Alderson, D.L., Funk, D., Gera, R., 2020. Analysis of the global maritime transportation system as a layered network. Journal of Transportation Security 13, 291–325.

[2] Álvarez, N.G., Adenso-Díaz, B., Calzada-Infante, L., 2021. Maritime traffic as a complex network: A systematic review. Networks and Spatial Economics 21, 387–417.

[3] Arakawa, A., Lamb, V.R., 1977. Computational design of the basic dynamical processes of the ucla general circulation model. Methods in Computational Physics: Advances in Research and Applications 17, 173–265. doi:https://doi.org/10.1016/B978-0-12-460817-7.50009-4.

[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.

[5] Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D., 2020. Language models are few-shot learners. arXiv arXiv:2005.14165.

[6] 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.

[7] Charalambopoulos, N., Xidias, E., Nearchou, A., 2023. Efficient ship weather routing using probabilistic roadmaps. Ocean Engineering 273, 114031. doi:https://doi.org/10.1016/j.oceaneng.2023.114031.

[8] Copernicus, . Copernicus: Europe's eyes on earth. https://www.copernicus.eu/es. Accessed on 10-07-2024.

[9] Dijkstra, E.W., et al., 1959. A note on two problems in connexion with graphs. Numerische mathematik 1, 269–271.

[10] Ducruet, C., 2020. The geography of maritime networks: A critical review. Journal of Transport Geography 88, 102824. doi:https://doi.org/10.1016/j.jtrangeo.2020.102824.

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

[12] E.U. Copernicus Marine Service Information (CMEMS), a. GLOBAL_ANALYSISFORECAST_PHY_001_024: Ocean/Sea Physical Analysis and Forecast. https://data.marine.copernicus.eu/product/GLOBAL_ANALYSISFORECAST_PHY_001_024/description. doi:https://doi.org/10.48670/moi-00016. accessed on 10-07-2024.

[13] E.U. Copernicus Marine Service Information (CMEMS), b. GLOBAL_ANALYSISFORECAST_WAV_001_027: Global Ocean Waves Analysis and Forecast. https://data.marine.copernicus.eu/product/GLOBAL_ANALYSISFORECAST_WAV_001_027/description. doi:https://doi.org/10.48670/moi-00017. accessed on 10-07-2024.

[14] European Centre for Medium-Range Weather Forecasts, 2016a. ECMWF IFS CY41r2 High-Resolution Operational Forecasts. doi:10.5065/D68050ZV.

[15] 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 on 10-07-2024.

[16] 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.

[17] 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 .

[18] 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. doi:10.1016/J.IFACOL.2021.11.097.

[19] Frasinaru, C., Raschip, M., 2019. Greedy best-first search for the optimal-size sorting network problem. Procedia Computer Science 159, 447–454. doi:10.1016/J.PROCS.2019.09.199.

[20] Freedman, D., Pisani, R., Purves, R., 2007. Statistics (international student edition). Pisani, R. Purves, 4th edn. WW Norton & Company, New York .

[21] Ge, J., Zhang, Q., Wan, Z., et al., 2022. Regional operating patterns of world container shipping network: A perspective from motif identification. Physica A: Statistical Mechanics and its Applications 607, 128171.

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

[23] Gleeson, E., Clancy, C., Zubiate, L., Janjić, J., Gallagher, S., Dias, F., 2019. Teleconnections and extreme ocean states in the northeast atlantic ocean. Advances in Science and Research 16, 11–29. URL: https://asr.copernicus.org/articles/16/11/2019/, doi:10.5194/asr-16-11-2019.

[24] Grandcolas, S., 2022. A metaheuristic algorithm for ship weather routing. Operations Research 3, 35.

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

[26] Hart, P.E., Nilsson, N.J., Raphael, B., 1968. A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics 4, 100–107. doi:10.1109/TSSC.1968.300136.

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

[28] 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. Quarterly Journal of the Royal Meteorological Society 146, 1999–2049. doi:https://doi.org/10.1002/qj.3803.

[29] Holtrop, J., Mennen, G., et al., 1982. An approximate power prediction method. International shipbuilding progress 29, 166–170.

[30] IMO, 2020. Fourth IMO GHG Study 2020 - doc. MEPC59/INF.10. International Maritime Organization (IMO), London, UK .

[31] Jiang, A.Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D.S., de las Casas, D., Hanna, E.B., Bressand, F., Lengyel, G., Bour, G., Lample, G., Lavaud, L.R., Saulnier, L., Lachaux, M.A., Stock, P., Subramanian, S., Yang, S., Antoniak, S., Scao, T.L., Gervet, T., Lavril, T., Wang, T., Lacroix, T., Sayed, W.E., 2024. Mixtral of experts. arXiv:2401.04088.

[32] Kim, M., Hizir, O., Turan, O., Day, S., Incecik, A., 2017. Estimation of added resistance and ship speed loss in a seaway. Ocean Engineering 141, 465–476.

[33] Kim, Y.R., Steen, S., 2022. Application of machine learning algorithms for predicting added resistance in arbitrary wave headings of a ship. Ocean Engineering Volume 5B: Ocean Engineering; Honoring Symposium for Professor Günther F. Clauss on Hydrodynamics and Ocean Engineering, V05BT06A026. doi:10.1115/OMAE2022-78433.

[34] Kuhlemann, S., Tierney, K., 2020. A genetic algorithm for finding realistic sea routes considering the weather. Journal of Heuristics 26, 801–825.

[35] 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.

[36] 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 Engineering 269, 113494. doi:https://doi.org/10.1016/j.oceaneng.2022.113494.

[37] 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. doi:10.5281/zenodo.3248739.

[38] Mannarini, G., Salinas, M.L., Carelli, L., Petacco, N., Orović, J., 2024. Visir-2: ship weather routing in python. Geoscientific Model Development 17, 4355–4382.

[39] McWilliams, J.C., 2000. Chapter 14 - formulation of oceanic general circulation models, in: Randall, D.A. (Ed.), General Circulation Model Development. Academic Press. volume 70 of *International Geophysics*, pp. 421–456. doi:https://doi.org/10.1016/S0074-6142(00)80062-5.

[40] 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. Applied Ocean Research 118, 102964. doi:https://doi.org/10.1016/j.apor.2021.102964.

[41] 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.

[42] 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 Science 16, 1385–1398. URL: https://os.copernicus.org/articles/16/1385/2020/, doi:10.5194/os-16-1385-2020.

[43] National Oceanic and Atmospheric Administration (NOAA), a. Beaufort wind scale. https://www.weather.gov/mfl/beaufort. Accessed on 10-07-2024.

[44] National Oceanic and Atmospheric Administration (NOAA), b. Estimating wave height using wind speed during a tropical cyclone. https://www.vos.noaa.gov/MWL/201512/waveheight.shtml. Accessed on 10-07-2024.

[45] National Oceanic and Atmospheric Administration (NOAA), c. Gulf stream speed. https://oceanservice.noaa.gov/facts/gulfstreamspeed.html. Accessed on 10-07-2024.

[46] NEMO TOP Working Group, 2022. TOP – Tracers in Ocean Paradigm – The NEMO Tracers engine. URL: https://doi.org/10.5281/zenodo.1471700, doi:10.5281/zenodo.1471700.

[47] Organization, I.M., 2017. Imo and sustainable development. URL: https://wwwcdn.imo.org/localresources/en/MediaCentre/HotTopics/Documents/IMO%20SDG%20Brochure.pdf.

[48] Organization, I.M., 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.

[49] P. Janssen, L. Aouf, A. Behrens, G. Korres, L. Cavalieri, K. Christiensen, O. Breivik, 2016. MyWave Project. URL: https://www.ecmwf.int/en/research/projects/mywave. accessed on 10-07-2024.

[50] Perera, L.P., Soares, C.G., 2017. Weather routing and safe ship handling in the future of shipping. Ocean Engineering 130, 684–695. doi:10.1016/j.oceaneng.2016.09.007.

[51] Polvani, L.M., Scott, R., Thomas, S., 2004. Numerically converged solutions of the global primitive equations for testing the dynamical core of atmospheric gcms. Monthly weather review 132, 2539–2552.

[52] 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.

[53] Precioso, D., Milson, R., Bu, L., Menchions, Y., Gómez-Ullate, D., 2024. Hybrid search method for zermelo's navigation problem. Computational and Applied Mathematics 43, 250.

[54] 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.

[55] Rinauro, B., Begovic, E., Mauro, F., Rosano, G., 2024. Regression analysis for container ships in the early design stage. Ocean Engineering 292, 116499.

[56] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G., 2023. Llama: Open and efficient foundation language models. arXiv:2302.13971.

[57] Townsin, R., Kwon, Y., 1983. Approximate formulae for the speed loss due to added resistance in wind and waves. Engineering, Environmental Science .

[58] 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. Maritime Business Review 6, 280–292.

[59] Uber Technologies, Inc., 2018. H3 - Hexagonal Hierarchical Geospatial Indexing System. Software .

[60] 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/.

[61] 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. URL: https://doi.org/10.5281/zenodo.7534900, doi:10.5281/zenodo.7534900.

[62] Vettor, R., Szlapczynska, J., Szlapczynski, R., Tycholiz, W., Soares, C.G., 2020. Towards improving optimised ship weather routing. Polish Maritime Research 27, 60–69.

[63] 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. Geoscientific Model Development 7, 663–693. URL: https://gmd.copernicus.org/articles/7/663/2014/, doi:10.5194/gmd-7-663-2014.

[64] 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. Journal of Marine Science and Engineering 12, 556. doi:10.3390/jmse12040556.

[65] Young-Joong, K., 2005. A research on the approximate formulae for the speed loss at sea. Journal of Ocean Engineering and Technology 19, 90–93.

[66] Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., Choi, Y., 2019. Hellaswag: Can a machine really finish your sentence? arXiv arXiv:1905.07830.

[67] 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. Journal of Ocean University of China 21, 28–38.

[68] Zis, T.P., Psaraftis, H.N., Ding, L., 2020. Ship weather routing: A taxonomy and survey. Ocean Engineering 213, 107697.

[69] Zyczkowski, M., Szlapczynski, R., 2023. Collision risk-informed weather routing for sailboats. Reliability Engineering & System Safety 232, 109015.

A. 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 $i = 0, 1, \dots, 2147$ and $j = 0, 1, \dots, 4319$, 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 [8]. We construct a bicubic polynomial like the following:

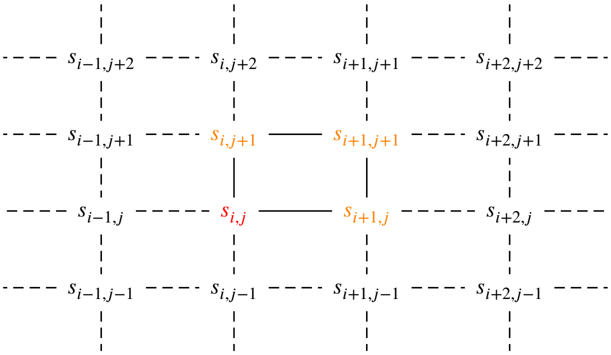
We build a bicubic polynomial for each $1/12° \times 1/12°$ square, this polynomial can be represented as:

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

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:

$$\mathbf{M} = \begin{bmatrix} \mathbf{0} & \mathbf{2A} & \mathbf{0} & \mathbf{0} \\ \mathbf{2B} & -\mathbf{A} & \mathbf{2A} & \mathbf{B} \\ \mathbf{A} & -\mathbf{2A} & \mathbf{A} & \mathbf{0} \\ \mathbf{B} & \mathbf{A} & -\mathbf{A} & -\mathbf{B} \end{bmatrix} \tag{18}$$

where

$$\mathbf{A} = \frac{1}{3} \begin{bmatrix} 0 & 6 & 0 & 0 \\ -2 & -3 & 6 & -2 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \tag{19}$$

and

$$\mathbf{B} = \frac{1}{9} \begin{bmatrix} 0 & -6 & 0 & 0 \\ 2 & 3 & -6 & 1 \\ -3 & 6 & -3 & 0 \\ 1 & -3 & 3 & -1 \end{bmatrix} \tag{20}$$

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

$$\begin{cases} f_{(i,j)}(-, 0) = f_{(i,j-1)}(-, 1) \\ f_{(i,j)}(-, 1) = f_{(i,j+1)}(-, 0) \end{cases} \tag{21}$$

$$\begin{cases} f_{(i,j)}(0, -) = f_{(i-1,j)}(1, -) \\ f_{(i,j)}(1, -) = f_{(i+1,j)}(0, -) \end{cases} \tag{22}$$

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:

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

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.

B. A⋆ hyper-parameter grid search

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

- Grid resolution: 3, 4, 5
- N-order neighbours: 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. 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 (circumnavigation), 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 neighbours 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.

We computed the Pearson correlation [20] between hyper-parameters and results, to better understand their impact. These Pearson coefficients are shown in Table 8, 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⋆ hyper-parameters, increasing the N-order neighbours significantly improves optimization, because it adds more nodes to explore at each step, as shown in Table **??**. Additionally, the weight of the heuristic and grid resolution greatly impact computation time, which is crucial for deployment and implementation of this system.

Table 8
Pearson correlation coefficient (PCC) between the gains produced by A⋆ (compared with the minimum distance), its computation time and the different hyper-parameters of this algorithm.

PCC	Gain	Compute time
Vessel speed	-0.258	0.030
***N*-order neighbours**	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

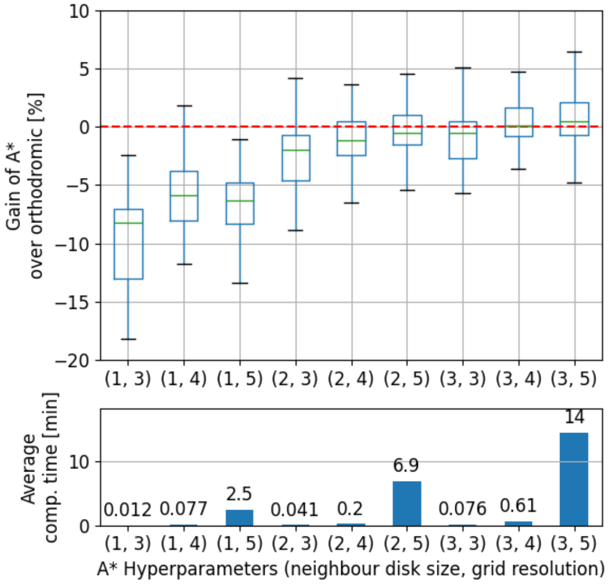


Figure 11: Hyper-parameter settings for A⋆, grouped by N-order neighbours and grid resolution. The gain of A⋆ over circumnavigation is shown, and computation times are plotted below.

Figure 11 groups the A⋆ gains by N-order neighbours and grid resolution, validating our expectations: a finer grid resolution and a bigger neighbour 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 neighbours. Next we decide a weight for the heuristic.

Looking at Table 9 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 5.2 to every A⋆ output. This ensures that FMS will always output a solution at least as good as the seed route provided by A⋆.

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

Table 9
Gain and computation time of A*, showing mean (std) across configurations. Grid resolution is fixed to 4, and 3rd-grade neighbours.

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)

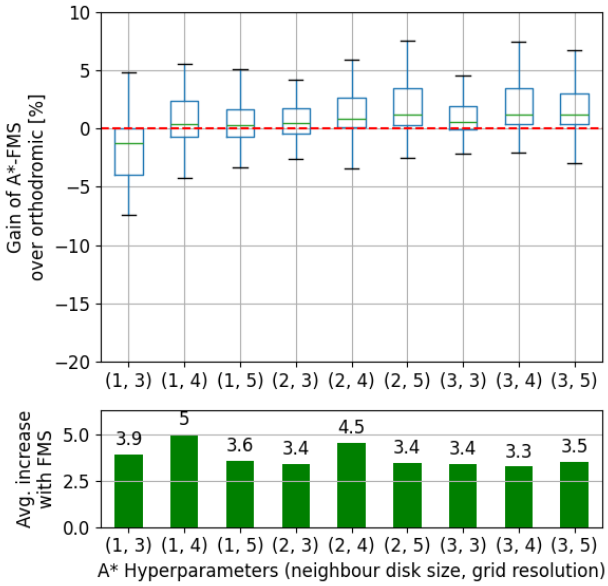


Figure 12: The box-plot show the gain of A*-FMS over circumnavigation, for every A* configuration. Below them, the bars show the average increase (in gain %) that FMS achieves when applied to A*.

FMS is able to improve any A* configuration greatly, as illustrated in Figure 12. 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 neighbours. The choice of A* hyper-parameters is not so crucial after applying the FMS algorithm, whose addition is a great improvement over a pure graph optimization method.