FOREX Publication
Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 1 | Pages 111-118 | e-ISSN: 2347-470X

A Route Planning Method using Neural Network and HIL Technology Applied for Cargo Ships

Huu-Khanh Doan[1*], Anh-Tuan Dinh[2] and Duc-Tuan Hoang[3]
[1,2,3]Faculty of Electrical – Electronics Engineering, Vietnam Maritime University, Hai Phong, Vietnam

***Correspondence:** Huu-Khanh Doan; khanhdh.ddt@vimaru.edu.vn

ABSTRACT- This paper presents the development of a method to find optimal routes for cargo ships with three criteria: fuel consumption, safety, and required time. Unlike most previous works, operational data are used for the studies. In this study, we use data collected from a hardware-in-loop (HIL) simulator, with the plant model being a 3D dynamic model of a bulk carrier designed and programmed from 6 degrees of freedom (6-DOF) equations that can interact with forces and moments from the environmental disturbances. The dataset generated from the HIL simulator with various operating scenarios is used to train an artificial neural network (ANN) model. This predictive model then combines the A* algorithm, weather forecast data, ship parameters, and waypoint coordinates to find the optimal routes for ships before each voyage. The test results show that the proposed method works reliably, helping to improve fuel efficiency and enhance the safety of the ships.

Keywords: route planning, A* algorithm, HIL simulator, artificial neural network

ARTICLE INFORMATION
Author(s): Huu-Khanh Doan, Anh-Tuan Dinh and Duc-Tuan Hoang;
Received: 20/12/2023; **Accepted**: 02/02/2024; **Published**: 05/02/2024;
e-ISSN: 2347-470X;
Paper Id: IJEER 2012-09;
Citation: 10.37391/IJEER.120116
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-12/ijeer-120116.html

crossref member
CROSSREF.ORG
THE CITATION LINKING BACKBONE

Publisher's Note: FOREX Publication stays neutral with regard to Jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

Shipping is the cheapest transportation method between countries and continents. Fuel prices are rising now, but fuel cost is the most expensive factor in operating a ship, accounting for up to 70% of the total cost of ship operation. Moreover, to comply with the International Convention for the Prevention of Pollution from Ships (MARPOL) of the International Maritime Organization (IMO) under Annex MEPC.328(76), revised in 2021, solutions to improve fuel efficiency are an issue that many researchers in this field are interested in.

There are many solutions to improve fuel efficiency for ships. Building optimal routes for ships before each voyage based on weather forecast data is a solution that brings high efficiency and low cost. Over the years, there have been many studies related to this issue.

The goal of the work [1] is to find the optimal route such that the main engine (ME) speed is the smallest. The authors use the Journée algorithm to estimate the travel time and the Newton-Raphson method to find the minimum ME speed. The study [2] uses a modified Dijkstra algorithm to find the shortest distance with the convention that the shortest distance is the distance the ship travels in the shortest time. The ship speed is estimated based on speed reduction curves.

The research [3] proposes to build a method to find the safe route for the ships based on Dijkstra and the genetic algorithm. The literature [4], [5] uses the Dijkstra algorithm to find the shortest path in a grid of nodes. The weights between any pair of nodes will be proportional to the distance between those two nodes and the added resistance to the ship. The work [6] builds software based on the Promethee method, which is a multicriteria decision support method. The criteria consist of fuel consumption, transit time, or environmental pollution. The study [7] uses an operational dataset recorded over a year of a 17,500 DWT cargo ship combined with the A* algorithm to find out the route with the smallest total propulsion forces. Just like in the literature [4], [5], the studies [8]–[10] also use the RAO to calculate the necessary weights used for the algorithms, such as dynamic programming algorithm in the works [9], [10] to find the route. The study [11] uses Dijkstra algorithm, and the cost function is calculated using the SPI index. The research [12] uses an algorithm that combines genetic algorithms and swarm optimization algorithms to find the route with the choice of three criteria: safety, fuel economy, and required time. The literature [13] uses the multi-objective ant colony algorithm to find the optimal route. Meanwhile, the studies [14]–[17] all use the A* algorithm, in which the research [14] uses a machine learning model from the data of the automatic identification system (AIS) combined with weather data to calculate travel speed, literature [15], [16] combine the A* algorithm and a wave prediction model to find the optimal route. In addition, the work [17] combines A* and a fuel consumption prediction model from the operational data.

The above studies show that many works have used Dijkstra and A* algorithms combined with other methods to find the cost function, such as using the RAO, genetic algorithms, dynamic programming algorithms, etc. However, in most of these works, the data are primarily historical operational data such as noon reports or AIS data. These data are often incomplete (mainly no data on current speed and current direction) or not large enough (for ships that have not been in service for a long time or newly

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 1 | Pages 111-118 | e-ISSN: 2347-470X

built ships), not to mention the large sampling time (usually once a day), using these incomplete datasets will reduce the accuracy and reliability of route finding algorithms.

Therefore, this study proposes a method to find optimal routes for ships with three criteria before each voyage. The proposed method uses the dataset generated from a HIL simulator, ANN model, and A* algorithm. Unlike most previous studies, the dataset created from the HIL simulator instead of past operating data fully includes the three main turbulence components affecting the ship: waves, wind, and current, which will help improve the accuracy and reliability of the route-finding algorithm. Furthermore, the novelty of this method is that it can be applied to ships where data is collected over a short time or some necessary data is missing, especially ships that are not authorized to use past operational data for security and defence reasons in Vietnam.

2. MATERIALS AND METHODS

2.1 Research Framework

Figure 1 displays the fundamental research framework. In *step 1*, we will build a HIL simulator based on the ship's technical profile, the 6-DOF equations of motion, and mathematical models of environmental disturbances and propulsion systems. A dataset similar to the operational data of the real ship with various operating scenarios will be generated from that HIL simulator. *Step 2* shows the construction of the proposed algorithms. The built ANN model combined with the coordinates of the starting waypoint, the destination waypoint, and the coordinates of the obstacles (if any) will be the input variables for *algorithm 1* to find the optimal fuel route. Besides, from the optimal found route and the required time, the appropriate ME speed will be recommended by *algorithm 2*.

2.2 Data Preparation

2.2.1. The HIL Simulator

This study uses a HIL simulator to build a dataset to replace operational datasets. In particular, the plant model built is a 3D dynamic model. Meanwhile, the controller is a trajectory controller built by MATLAB/Simulink software. The 3D dynamic model is built based on the 6-DOF equations of motion and the ship's technical profile. These equations are commonly used in ship motion control studies as follows [18]:

$$\begin{aligned} \dot{\eta} &= J(\eta)\upsilon \\ M\dot{\upsilon} + C(\upsilon)\upsilon + D(\upsilon) + G\eta &= \tau_E + \tau \end{aligned} \qquad (1)$$

where η denotes the position and orientation vector; $J(\eta)$ is a transformation matrix; υ is the body-fixed linear and angular velocity vector; $M = M_{RB} + M_A$; $C(\upsilon) = C_{RB}(\upsilon) + C_A(\upsilon)$ with M_{RB} is inertia matrix, M_A is added inertia matrix, $C_{RB}(\upsilon)$ is Coriolis and Centripetal matrix, $C_A(\upsilon)$ is the hydrodynamic Coriolis and centripetal matrix, G is a constant matrix, $D(\upsilon)$ is the damping matrix, τ_E and τ are environmental and propulsion forces and moments, respectively.

To build the 3D dynamic ship, we use Unity software, which is specialized software in 3D graphics simulation and processing. The shape and dimensions of the real ship (the real ship used in this study is a bulk carrier named The Prosperity) are designed into a 3D model as an FBX file to be imported into the Unity software. The 3D ship will then be assigned the Rigidbody property so that it can have physical interactions with its surroundings. In addition, the software modules (C# language) are also programmed to assign corresponding objects. More details on the construction and testing of this model can be found in the work [19].

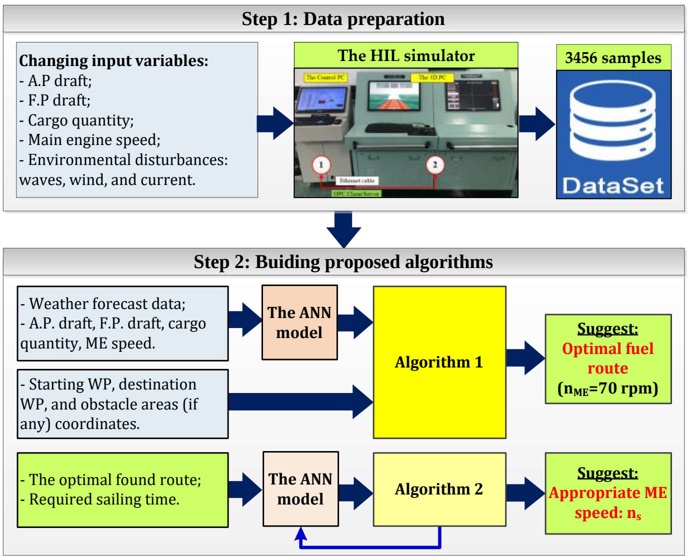

Figure 1: Research framework

When building the HIL simulator, in addition to the 3D dynamic model, a real-time fuel consumption calculation model is also built based on the interpolation method and semi-empirical formulas. This model has been presented in detail in research [20].

Figure 2 shows the built HIL simulator. *Block 1* contains the Control PC, which acts as an actual controller. Meanwhile, *Block 2* consists of the 3D PC running a 3D dynamic model as a plant model. These two blocks are communicated with each other *via* an ethernet cable with the OPC Client/Server protocol.

2.2.2. Data Preparation

Firstly, we make the following assumptions to limit the scope of the study:

- The ship's draft is 14.429 m (laden);
- The ship is on an even keel (the draft of the ship is equal fore and aft);
- Waves are generated entirely by the wind;
- Wind speed and wave height are programmed according to the Beaufort scale of the World Meteorological Organization (WMO) [21].

The input variables will be varied to create different scenarios to build the dataset. Specifically, there are five input variables,

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 1 | Pages 111-118 | e-ISSN: 2347-470X

including the ME speed (03 values: 65, 70, and 75 rpm), wind speed (04 values: 0, 9, 31, and 45 knots), wind direction (12 values: 0:30:330 degrees), current speed (03 values: 0, 1.5, 3 knots), and current direction (08 values: 0:45:315 knots). Meanwhile, the output variables include fuel consumption and sailing time. Therefore, we get a dataset with 3456 cases. In all test cases, the ship controlled by a PID controller goes through the same sample route.

2.3 The Proposed Algorithms

2.3.1. A* Search Algorithm

A* is a computer science algorithm commonly used in path-finding and graph traversal. As A* crosses the map, it follows a track of the lowest known heuristic cost, keeping an arranged priority queue of consecutive track nodes along the path [22]. The basic principle of this method depends on the following equation:

$$f(x)=g(x)+h(x) \tag{2}$$

Where x is the current node on the map, and $g(x)$ is the actual cost from the starting node to the current node x. Meanwhile, $h(x)$ is the heuristic function that estimates the cost from the current node x to the destination node, and $h(x)$ is admissible if ∀x:

$$h(x) \le h^*(x) \tag{3}$$

Where $h^*(x)$ is the actual cost from node x to the destination node.

Thus, before building algorithms to find the optimal route for the ship, we need to do the following:

- Build a grid as a basis for implementing algorithms and calculate the estimated cost matrix H;
- Calculate the actual cost matrix G by building the ANN model;
- Build the objective function;

2.3.2. Building the Curved Grid

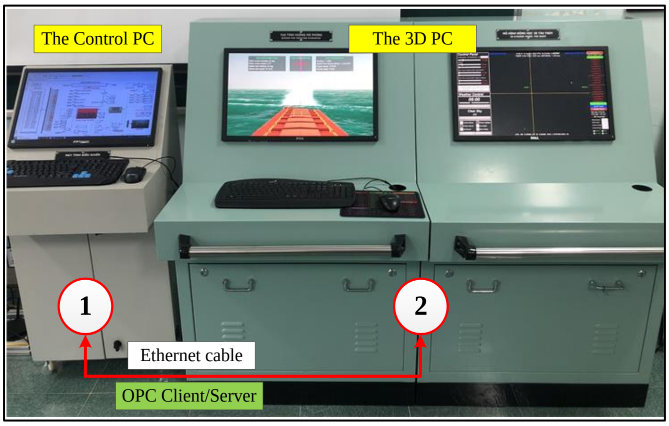

Figure 2: The built HIL simulator

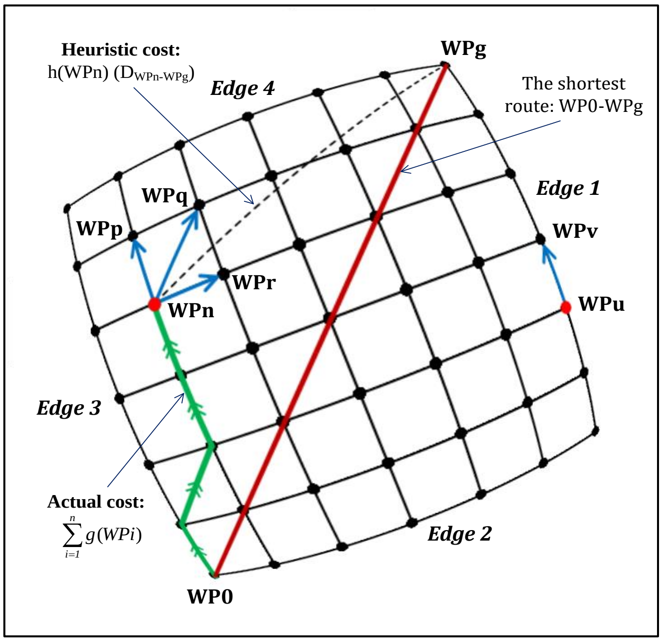

Figure 3: The curved grid

Like the [2], [4] studies, to implement the A* algorithm, it is first necessary to create a moving graph for the ship. We know that the shortest distance between any two waypoints on Earth is an arc on the great circle. Therefore, based on the coordinates of starting waypoint WP0 and destination waypoint WPg, this study proposes to build a moving graph for ships that is a curved grid, as shown in *figure 3*. For any waypoint like WPn, there are three directions to travel from WPn to WPp, WPq, or WPr. Meanwhile, with waypoints located on Edge 1 and Edge 4, such as WPu, there is only one direction of travel from WPu to WPv. In particular, the shortest distance between any two waypoints is calculated according to the Haversian formula. For example, the shortest arc length $D_{WPn-WPq}$ between any two points, WPn and WPq, will be calculated using *equation (4)*.

$$D_{WPn-WPq} = R.c \tag{4}$$

Where R is Earth's radius and c is a coefficient calculated as the following equations:

$$\begin{aligned} c &= 2.a\tan(\sqrt{a},\sqrt{(1-a)}) \\ a &= \sin^2(\Delta\varphi/2)+\cos\varphi_1.\cos\varphi_2.\sin^2(\Delta\lambda/2) \end{aligned} \tag{5}$$

Where $\Delta\varphi=\varphi_{WPn}-\varphi_{WPq};\Delta\lambda=\lambda_{WPn}-\lambda_{WPq}$; $(\varphi_{WPn},\lambda_{WPn})$ are longitude and latitude of WPn; and $(\varphi_{WPq},\lambda_{WPq})$ are longitude and latitude of WPq.

Therefore, the estimated cost when the ship moves from WPn to WPq will be calculated as follows:

$$H_{WPn-WPq} = D_{WPn-WPq} \times FC_{unit} \tag{6}$$

Where *FC unit* is the fuel consumed per nautical mile in calm sea conditions to satisfy *equation (3).*

2.3.3. The ANN Model for Predicting Fuel Consumption and Sailing Time

The proposed prediction model is a multilayer perceptron (MLP) trained by the back-propagation method with five inputs and two outputs corresponding to the number of inputs and outputs of the built dataset. According to many studies, the number of hidden layers and neurons of the hidden layer will depend on each different dataset. Therefore, the authors trained the neural network with 30 models with different numbers of hidden layers and neurons with training parameters shown in *table 1*. When testing trained 30 models with the testing dataset, the network model with one hidden layer and 30 neurons in that hidden layer is the model with the smallest mean absolute percentage error (MAPE) of only 1.031%. The structure of this ANN model contains one input layer (five neurons), one hidden layer (30 neurons), and one output layer (two neurons), as shown in *figure 4.*

Table 1: Main parameters for training the ANN model

Training parameters	Values
Training function	Levenberg-Marquardt backpropagation
The transfer function of the input layer and hidden layer	tansig
The transfer function of the output layer	purelin
Loss function	Mean squared error
Learning rate	0.001
Maximum training epochs	5000
Performance goal	0.3
The number of hidden layers	1 ÷ 3 (interval: 1 layer)
The number of neurons in hidden layers	10 ÷ 100 (interval: 10 neurons)

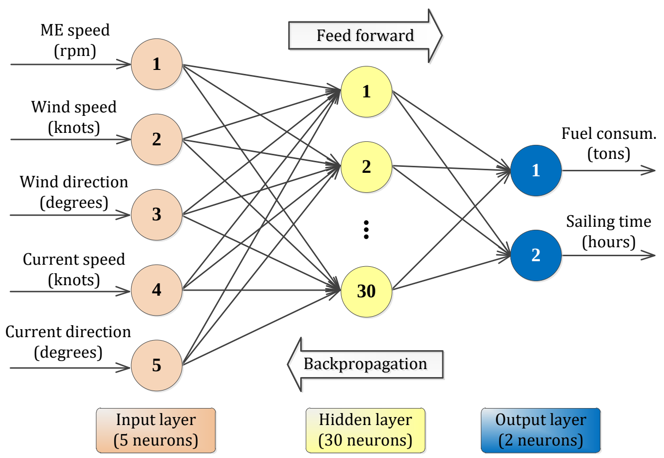

Figure 4: Structure of the ANN model

More details about this model, the HIL simulator, and the generated dataset are presented in the research [23].

2.3.4. The Objective Function

The proposed algorithm will minimize the objective function as follows:

$$Minimize: f(WPn) = h(WPn) + \sum_{i=1}^{n} g(WPi) \tag{7}$$

Where *h(WPn)* is the estimated fuel consumption to go from WPn to WPg, and *g(WPi)* is the actual fuel consumption to reach waypoint (i) from its parent waypoint (i-1).

The *h(WPn)* values will be taken from the matrix H, which will be calculated based on the curved grid and Harvesin formula as *equation (6)*. Meanwhile, the *g(WPi)* values will be taken from the matrix G, which will be calculated using the proposed ANN model, the weather forecast data, and the distance between adjacent waypoints.

2.3.5. The proposed algorithms

This section proposes two algorithms to create an optimal route with three criteria: *fuel economy, safety and required time*. *algorithm 1* will find the optimal fuel route based on the A* algorithm and ANN model with the ME speed of 70 rpm. Initially, the OPEN and CLOSED list will be created. The OPEN list is used to record all waypoints that need to be considered to find the optimal route. Meanwhile, the CLOSED list is a list that stores waypoints that we no longer need to review. *Algorithm 1* will start by finding neighboring points with WP_0 until the last neighboring point WPg to satisfy *equation (7).*

Specifically, *algorithm 1* will execute from line 1 to line 24. Initially, the OPEN list only includes WP_0, so WPn==WP_0 and WP0 will be added to the CLOSED list. Then, *algorithm 1* will find all neighbors WPx of WPn (line 8). If WPx belongs to the OPEN list or CLOSED list, it will proceed from line 10 to line 15. Meanwhile, if WPx does not belong to either of the lists above, it will be added to the OPEN list (line 17). For each such WPx, the algorithm will calculate the actual fuel consumption g(WPx) based on matrix G (line 19) and the estimated fuel consumption h(WPx) based on matrix H (line 20), as mentioned in *section 2.3.4*.

In addition, in the case of WPx located in dangerous areas with a risk of collision, g(WPx) will be assigned to infinity. Therefore, *algorithm 1* will find routes that are both optimal for fuel consumption and safe, avoiding the risk of collision, if any. Details of *algorithm 1* are shown as pseudocodes in *table 2*.

Table 2: Algorithm 1

```
Initial:
+ OPEN={WP0}; CLOSED= ∅; g(WP0)=0;
+ h(WP0)=heu_cost(WP0,WPg);
+ f(WP0)=h(WP0)+g(WP0);
1. while OPEN ≠ ∅
2. find WPn with the least f(WPn) in the OPEN;
3. if (WPn == WPg)
4. return "Route is found";
5. else
6. remove WPn from OPEN; add WPn to CLOSED;
7. endif
```

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 1 | Pages 111-118 | e-ISSN: 2347-470X

8. for WPx ∈ neighbors(WPn) do
9. current_cost = g(WPn)+cost(WPn,WPx);
10. if (WPx ∈ OPEN & g(WPx) > current _cost)
11. remove WPx from OPEN;
12. endif;
13. if (WPx ∈ CLOSE & g(WPx) > current _cost)
14. remove WPx from CLOSE;
15. endif;
16. if (WPx ∉ OPEN & WPx ∉ CLOSED)
17. add WPx to OPEN;
18. g(WPx) = current_cost;
19. h(WPx) =heu_cost(WPx,WPg);
20. f(WPx) = g(WPx) + h(WPx);
21. endif;
22. endfor
23. endwhile;
24. return "Route cannot be found";

In fact, the time required for the ship to move from WP_0 to WP_g may be larger or smaller than the time it takes for the ship to move from WP_0 to WP_g on the route found by *algorithm 1* with the ME speed of 70 rpm. Therefore, *algorithm 2* is proposed to find a suitable speed for the ME so that the ship can arrive at WPg on time. Details of *algorithm 2* are shown as pseudocodes in *table 3*. First, the ANN model will be used to predict the sailing time t_i when the ship moves from WP_0 to WP_g, corresponding to the ME speed of 70 rpm (line 1). If the required sailing time t_r is greater or less than t_i then *algorithm 2* will decrease or increase n_i by 0.1 rpm at each iteration until $|\Delta t| \le$ DB (deadband value) will find the appropriate ME speed ns (line 3 to line 13).

Table 3: Algorithm 2

Initial:
+ The optimal routes just found by *Algorithm 1*;
+ $n_i = n_d$ (70 rpm);
1. using the ANN model to predict the sailing time t_i for the optimal route just found corresponding to n_i;
2. calculate the $\Delta t = t_r - t_i$; *% t_r is the required time*
3. if $|\Delta t|$ > DB; *% DB is the deadband value*
4. if $\Delta t > 0$
5. $n_i = n_i - 0.1$;
6. goto 1;
7. else
8. $n_i = n_i + 0.1$;
9. goto 1;
10. endif
11. else
12. get the suggested $n_s = n_i$;
13. endif

3. RESULTS AND DISCUSSION

3.1. Testing scenarios

We assume the distance between the starting and destination waypoints is 11.314 nm (20953 m) for testing purposes. The middle of them will be divided into a grid (9 × 9) with 81 waypoints (the resolution of the grid will be 1 nm), and the weather conditions are assumed, as shown in *figure 5*. Because the test area is very small when compared to the Earth's radius, the curved grid is almost a square grid. There are three scenarios given as follows:

• ***Scenario 1*:** Waves and winds are only in the areas shown in *figure 5*. Red arrows represent wave and wind directions. Wind speed and wave height will correspond to the Beaufort scale of the WMO placed on the right. For example, the pink area in the middle of the grid corresponds to the wind speed of about 41÷47 knots and wave heights of about 7÷10 m (number 9/12 Beaufort).

• ***Scenario 2*:** There is an additional influence of current besides waves and winds, like in *Scenario 1*. Current is assumed to be in the blue dashed rectangular area, with blue arrows showing the current direction, assuming the current speed is 2 knots in this whole area.

• ***Scenario 3*:** This scenario is basically the same as *Scenario 2*, but it has obstacles. Specifically, waypoints 53, 54, 62, and 63 are dangerous areas where collisions can occur.

3.2. Results

With a pre-computed H matrix, the G matrix is also calculated after the weather data of all waypoints is available. After running *algorithm 1* for *Scenario 1*, the results are shown in *figure 6*. *Algorithm 1* recommends the route through waypoints 1-10-19-28-38-48-58-59-60-61-71-81 (Route 1). For *Scenario 2*, *algorithm 1* proposes the route through waypoints 1-2-3-13-14-24-34-44-54-63-72-81 (Route 3), as shown in *figure 7*.

Finally, with the same weather scenario as *Scenario 2* but with obstacles, *algorithm 1* recommends Route 6 through waypoints 1-2-3-13-14-24-34-43-52-61-71-81, as shown in *figure 8*.

For *algorithm 2*, six cases to find the appropriate speeds for three routes suggested by *algorithm 1* in three different scenarios will be tested. Each case will assume two different required time values (t_r) greater and less than the default sailing time (t_d) at the speed of 70 rpm. *Algorithm 2* gives results with suggested ME speeds (n_s) as in *table 5*.

After obtaining the recommended routes and ME speeds from the proposed algorithms, a trajectory controller based on fuzzy logic and a PID controller is built to verify the results [24]. This controller is installed on the Control PC. The results of verifying *Algorithm 1* are shown in *table 4* and *figure 9*. Meanwhile, *table 5* shows the results of verifying *algorithm 2*.

3.3. Discussion

With *Scenario 1*, when following the recommended route (Route 1), the ship saves fuel only 0.83% but is safer than running through the shortest route with heavy weather (Route 2 – great circle). Although the ship has to travel a greater distance of 14.56% (Route 1: 13.14 nm; Route 2: 11.47 nm), the resulting sailing time is also greater.

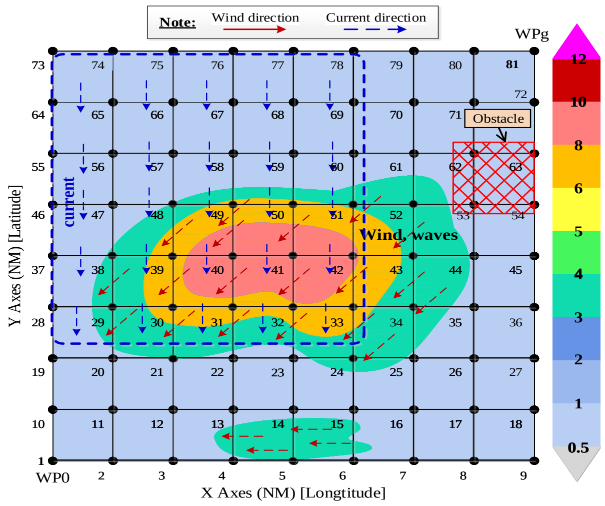

Figure 5: Testing conditions

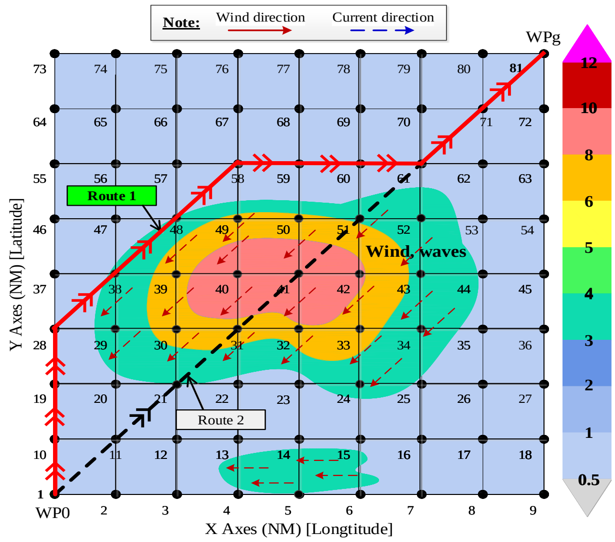

Figure 6: Recommended Route 1 (Scen. 1)

In *Scenario 2*, the recommended route is Route 3, which saves 4.01% of fuel compared to the case where the ship sails through the shortest route - Route 4 (similar to Route 2, but Route 4 has the effect of current). If the ship follows Route 5 as suggested in *Scenario 1*, it will consume 2.52% more fuel than Route 3 (Route 5 is similar to Route 1 but has the effect of current).

Scenario 3 has weather conditions like *Scenario 2*, but some waypoints are not allowed to enter (waypoints 53, 54, 62, and 63). *Algorithm 1* suggests Route 6. This route saves 3.97% of fuel compared to the case where the ship sails through Route 4, but this route consumes more fuel when running through Route 3 by 0.045%. The extra cost compared to Route 3 is minimal while the ship has sailed through a safer route, avoiding the risk of a dangerous collision.

Table 4 also shows that the largest error between the fuel consumption forecast data by the ANN model and the data when running the 3D model is 2.26% with a heavy sea. Compared with fuel forecasting models in other works, such as work [25], this error is not large and completely acceptable.

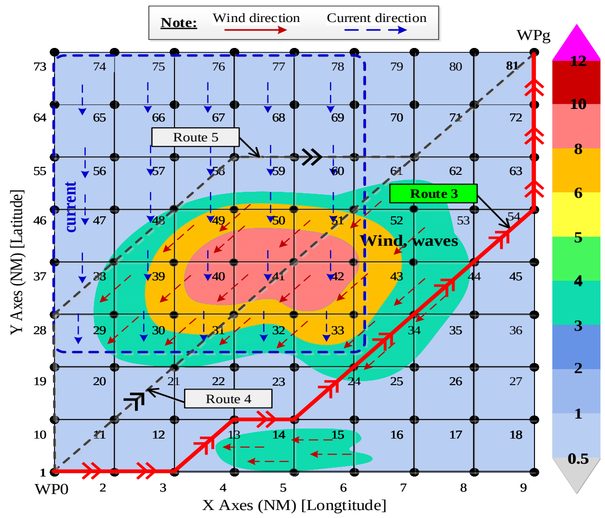

Figure 7: Recommended Route 3 (Scen. 2)

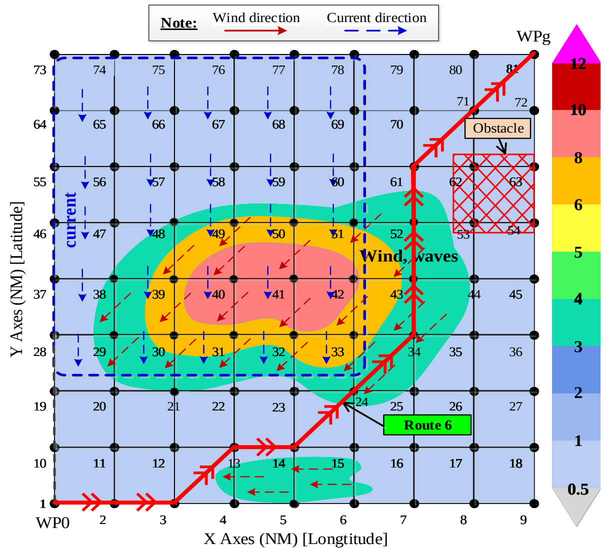

Figure 8: Recommended Route 6 (Scen. 3)

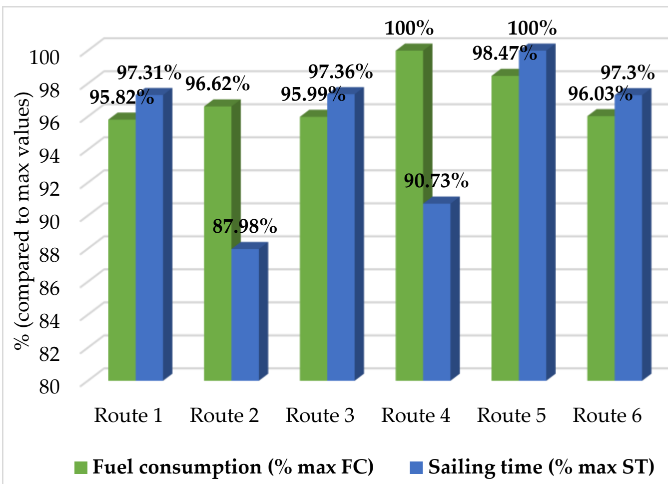

Figure 9: Fuel consumption and sailing time when compared with the corresponding max values for all six routes

From *table 5*, it can be seen that in each case, if the required time t_r is greater than the time t_d, *algorithm 2* will give a suggested speed that is smaller than the default speed to save as much fuel as possible for the ship. Conversely, if t_r is shorter than t_d, *algorithm 2* recommends a larger speed so that the ship can reach the destination on time. In addition, the largest error between t_r and actual sailing time ta when running the 3D dynamic model in all 6 cases is only 0.502%. These results show that *algorithm 2* works reliably.

Table 4: Comparison between the predicted data and the data obtained from the HIL simulator

Route	3D model Dist. (nm)	SOG (knots)	Fuel consumption (FC) (kgs) Predict	3D model	Error (%)	Sailing time (ST) (seconds) Predict	3D model	Error (%)	Fuel saving (%)
Route 1	13.14	12.18	810.56	805.61	0.61	3908.64	3886.5	0.57	0.83
Route 2	11.47	11.75	830.67	812.35	2.26	3567.34	3513.75	1.53	-
Route 3	13.14	12.16	813.7	807.05	0.83	3913.68	3888.25	0.65	4.01
Route 4	11.5	11.42	855.48	840.77	1.75	3664.79	3623.5	1.14	-
Route 5	13.15	11.85	833.81	827.87	0.72	4020.31	3993.75	0.67	-
Route 6	13.134	12.17	816.36	807.41	1.11	3912.86	3885.75	0.7	3.97

Table 5: Results when testing *Algorithm 2* with two values of tr

Route	t_d (s) [70 rpm]	t_r (s)	n_s (rpm)	t_a (s)	Error (%)
Route 1	3886.5	3780s	72.3	3761	**-0.502**
		4200s	64.4	4221	0.5
Route 3	3888.25	3780s	72.4	3763	-0.449
		4200s	64.5	4219	0.452
Route 6	3885.75	3780s	72.3	3763	-0.449
		4200s	64.5	4216	**0.381**

When comparing the results with other documents with the same research direction as this paper, typically documents [12], [13], [15], and [17]. These studies all use historical ship operating datasets and lack the influence of current. Assuming that *Scenario 2* takes into account the influence of current (both direction and magnitude), the algorithm finds the optimal fuel route with a combination of the A* algorithm and the neural network (as used in the research [17]) recommends Route 3. However, if the influence of current is ignored, the algorithm will recommend Route 5, the same as Route 1 in *Scenario 1*. Specific comparison results are shown in the following table:

Table 6: The suggested route considers and does not consider the influence of current

Studies	Recom. routes	FC (kgs)	ST(s)	Saving (%)
This method (with current)	Route 3	807.05	3888.25	2.58%
Other studies (no current)	Route 5	827.87	3993.75	-

The results in *table 6* show that with the same route-finding algorithm, if all disturbances affecting the ship are taken into account, it can be 2.58% more effective than other studies on finding optimal fuel routes when using incomplete datasets.

Thus, we see that *algorithm 1* will find routes to ensure two criteria: fuel economy and safety. Meanwhile, based on the route found by *algorithm 1, algorithm 2* will help find a route that meets the required time criterion (the ship can reach the destination on time as required).

4. CONCLUSIONS

This article has built a method to find routes for ships with three criteria: fuel economy, safety, and required time. Algorithm testing results show that the algorithm works effectively with low errors. The route proposed by *algorithm 1* can save up to 4.02% fuel for the ship when tested in *scenario 2*. In addition, *algorithm 2* can suggest the set speed values for ME to help the ship reach the destination on time with the optimal fuel routes found by *algorithm 1* with the largest error of only 0.502%.

The proposed method uses a dataset generated from a HIL simulator with all the main disturbance components that will help improve the accuracy and reliability of the route-finding algorithm. In addition, this method can also be applied to ships with limited or no access to operational data, such as security and defense reasons.

In the future, the algorithms will be tested on a larger scale of testing areas and evaluate the effectiveness of the proposed routes with the real ship's past routes before applying it to cargo ships and other types of ships in Vietnam to improve transportation efficiency.

Acknowledgments
This research is funded by Vietnam Maritime University.

REFERENCES

[1] C. de Wit, "Proposal for Low Cost Ocean Weather Routeing," J. Navig., vol. 43, no. 3, pp. 428–439, 1990, doi: 10.1017/S0373463300014053.

[2] A. Anel, "optimum track ship routing NAVAL POSTGRADUATE," 2005.

[3] J. Bekker and J. Schmid, "Planning the safe transit of a ship through a mapped minefield," ORiON, vol. 22, no. 1, 2006, doi: 10.5784/22-1-30.

[4] C. P. Padhy, D. Sen, and P. K. Bhaskaran, "Application of wave model for weather routing of ships in the North Indian Ocean," Nat. Hazards, vol. 44, no. 3, pp. 373–385, 2008, doi: 10.1007/s11069-007-9126-1.

[5] D. Sen and C. P. Padhy, "An approach for development of a ship routing algorithm for application in the North Indian Ocean region," Appl. Ocean Res., vol. 50, pp. 173–191, 2015, doi: 10.1016/j.apor.2015.01.019.

[6] C. Diakaki et al., "A decision support system for the development of voyage and maintenance plans for ships," Int. J. Decis. Support Syst., vol. 1, no. 1, p. 42, 2015, doi: 10.1504/ijdss.2015.067274.

[7] M. Bentin, D. Zastrau, M. Schlaak, D. Freye, R. Elsner, and S. Kotzur, "A New Routing Optimization Tool-influence of Wind and Waves on Fuel Consumption of Ships with and without Wind Assisted Ship Propulsion Systems," Transp. Res. Procedia, vol. 14, pp. 153–162, 2016, doi: 10.1016/j.trpro.2016.05.051.

[8] R. Vettor and C. Guedes Soares, "Development of a ship weather routing system," Ocean Eng., vol. 123, pp. 1–14, 2016, doi: 10.1016/j.oceaneng.2016.06.035.

[9] R. Zaccone, M. Figari, M. Altosole, and E. Ottaviani, "Fuel saving-oriented 3D dynamic programming for weather routing applications," Proc. 3rd Int. Conf. Marit. Technol. Eng. MARTECH 2016, vol. 1, no. June 2016, pp. 183–192, 2016, doi: 10.1201/b21890-26.

[10] R. Zaccone, M. Figari, and M. Martelli, "An optimization tool for ship route planning in real weather scenarios," Proc. Int. Offshore Polar Eng. Conf., vol. 2018-June, no. February 2019, pp. 738–744, 2018.

[11] S. Pennino, S. Gaglione, A. Innac, V. Piscopo, and A. Scamardella, "Development of a new ship adaptive weather routing model based on seakeeping analysis and optimization," J. Mar. Sci. Eng., vol. 8, no. 4, 2020, doi: 10.3390/JMSE8040270.

[12] W. Zhao, Y. Wang, Z. Zhang, and H. Wang, "Multicriteria ship route planning method based on improved particle swarm optimization–genetic algorithm," J. Mar. Sci. Eng., vol. 9, no. 4, 2021, doi: 10.3390/jmse9040357.

[13] J. Yang, L. Wu, and J. Zheng, "Multi-Objective Weather Routing Algorithm for Ships: The Perspective of Shipping Company's Navigation Strategy," J. Mar. Sci. Eng., vol. 10, no. 9, 2022, doi: 10.3390/jmse10091212.

[14] Y. W. Shin et al., "Near-optimal weather routing by using improved A* algorithm," Appl. Sci., vol. 10, no. 17, pp. 1–23, 2020, doi: 10.3390/app10176010.

[15] X. Jin, J. Xiong, D. Gu, C. Yi, and Y. Jiang, "Research on Ship Route Planning Method Based on Neural Network Wave Data Forecast," IOP Conf. Ser. Earth Environ. Sci., vol. 638, no. 1, 2021, doi: 10.1088/1755-1315/638/1/012033.

[16] M. Grifoll, C. Borén, and M. Castells-Sanabra, "A comprehensive ship weather routing system using CMEMS products and A* algorithm," Ocean Eng., vol. 255, no. May, 2022, doi: 10.1016/j.oceaneng.2022.111427.

[17] Y. Li, J. Cui, X. Zhang, and X. Yang, "A Ship Route Planning Method under the Sailing Time Constraint," J. Mar. Sci. Eng., vol. 11, no. 6, p. 1242, 2023, doi: 10.3390/jmse11061242.

[18] T.I. Fossen, Marine Control Systems, vol. 3, no. April. 2002.

[19] K. D. Huu, T. D. Anh, and T. H. Duc, "A 3D dynamic model applied for cargo ships to study ship motion control,"

[20] K. D. Huu, T. D. Anh, and T. H. Duc, "A Real-Time Model Using Interpolation Method and Semi-Empirical Formulas to Estimate Fuel Consumption for Cargo Ships," Proc. - 2023 Int. Conf. Ind. Eng. Appl. Manuf. ICIEAM 2023, pp. 932–937, 2023, doi: 10.1109/ICIEAM57311.2023.10139155.

[21] W. M. Organization, Guide to Wave Analysis and Forecasting, vol. 1998, no. 702. 1998.

[22] I. M. Zidane and K. Ibrahim, "Wavefront and a-star algorithms for mobile robot path planning," Adv. Intell. Syst. Comput., vol. 639, no. May, pp. 69–80, 2018, doi: 10.1007/978-3-319-64861-3_7.

[23] K. D. Huu, T. D. Anh, and T. H. Duc, "A Neural Network-Based Model to Predict Fuel Consumption and Sailing Time for Cargo Ships," Proc. - 2023 Int. Russ. Autom. Conf. RusAutoCon 2023, pp. 386–391, 2023, doi: 10.1109/RusAutoCon58002.2023.10272784.

[24] Y. A. Ahmed and K. Hasegawa, "Fuzzy Reasoned Waypoint Controller for Automatic Ship Guidance," in IFAC-PapersOnLine, Elsevier B.V., 2016, pp. 604–609. doi: 10.1016/j.ifacol.2016.10.501.

[25] L. T. Le, G. Lee, K. S. Park, and H. Kim, "Neural network-based fuel consumption estimation for container ships in Korea," Marit. Policy Manag., vol. 47, no. 5, pp. 615–632, 2020, doi: 10.1080/03088839.2020.1729437.