Different strategies to improve isochrone voyage optimization algorithm

Citation for the original published paper (version of record):

Chen, Y., Mao, W., Zhang, C. (2023). Different strategies to improve isochrone voyage optimization algorithm. Advances in the Analysis and Design of Marine Structures - Proceedings of the 9th International Conference on Marine Structures (MARSTRUCT 2023): 53-61.
http://dx.doi.org/10.1201/9781003399759-6

Different strategies to improve isochrone voyage optimization algorithm

Yuhan Chen & Wengang Mao
Department of Mechanics and Maritime Sciences, Chalmers University of Technology, Gothenburg, Sweden

Chi Zhang
National Center for Waterborne Transport Safety, Wuhan University of Technology, Wuhan Hubei, China

ABSTRACT: Voyage optimization can be essential in ship routing for autonomous and intelligent operations. Various optimization algorithms were proposed to search for optimal ship routing with minimized fuel consumption and accurate arrival time. In this paper, the Isochrone method, which is well-known for its robustness and efficiency, is further improved by different strategies to inherit the original capability of computation efficiency, and overcome its incompetence of multi-objective optimization, reliable convergence for approaching destination and feasibility for practical navigation. Especially an innovative way of searching at a late stage is proposed, as well as a flexible evaluation of cost functions according to different optimization objectives. The improved Isochrone method can fast optimize a shipping route with multi-objective purposes, such as accurate expected time of arrival and minimum fuel consumption. The effectiveness and efficiency of different improved Isochrone methods are verified by several North Atlantic voyages collected by a chemical tanker.

1 INTRODUCTION

A voyage planning system as an essential solution for e-navigation can be implemented to reduce air emissions from shipping (DNV 2014) and assist autonomous ship navigation. A proper optimization algorithm is a crucial component of a ship's voyage planning to achieve specific predefined objectives, such as minimum fuel cost, accurate time of arrival, etc. (Wang et al. 2019). The voyage optimization algorithms can be categorized into deterministic algorithms and stochastic algorithms (Wang et al. 2021). For the voyage planning systems available in the shipping market, deterministic algorithms are widely used due to their efficiency of optimization. In those methods, a ship's voyage is divided into several time stages. Dependent on how the searching area along the voyage in time is discretized, the optimization algorithms can be divided into dynamic and static grid-based voyage optimizations. The isochrone ideas first proposed by James (1957) and implemented by Hagiwara (1989) for minimum time voyage and the Dividing RECTangles algorithm, seen Larsson et al. (2015), are two typical dynamic grid searching methods. The static grid-based methods have predefined searching waypoints to choose from. If a ship's speed is fixed along the voyage, the methods are recognized as 2D voyage optimization methods, such as the dynamic programming method proposed by Bellsman (1952) and utilized to develop voyage optimization systems as in Chen (1978), de Wit (1990) and Calvert et al. (1991) to optimize a ship's route/path with fixed speed or power setting. In the static grid methods, if a ship's speed/power is configured to vary along the voyage, it is recognized as 3D voyage optimization method. For example, the 2D dynamic programming method was further developed by Shao et al. (2012), Zaccone et al. (2017 & 2018), and Wang et al. (2019) to allow voluntary speed reduction to solve three-dimensional voyage optimization problems.

However, voyage optimization should consider the balance between computation efficiency and the effectiveness of optimization algorithms for practical voyage planning. Sophisticated methods such as 3D deterministic and stochastic optimization algorithms are generally too complicated to solve voyage optimization problems in a reasonable time for arbitrary voyages because of many variables and their ambiguous dependencies. According to the market survey by Simonsen et al. (2015), shipping companies would like to complete a voyage optimization in less than 1 minute, preferably within 15 seconds. Concerning the fact that large amounts of computation efforts are spent in extracting sea environmental data for all discretized waypoints (spatial and temporal) during the optimization process, in this paper, the original Isochrone voyage optimization method is further developed to not only overcome the cons in the original Isochrone method but also provide more robust optimization results in a computation efficient manner. The effectiveness and efficiency of the improved method will be verified by full-scale measurement data from a chemical tanker.

2 OVERVIEW OF ORIGINAL ISOCHRONE ALGORITHM FOR VOYAGE OPTIMIZATION

For a ship voyage planning system, some sailing constraints, e.g., land avoidance, no-go zones, traffic separation scheme, etc., should be defined first. Then, the core elements of such a system are optimization algorithms. This study aims to develop original Isochrone method further to allow for fast and practical voyage planning. The Isochrones mean lines that the ship can reach with equal sailing time. The Isochrone voyage optimization method was initially proposed by James (1957) for manual use by navigators to help ships reach a destination as soon as possible (or on the exact estimated time of arrival (ETA)). For each time stage in the method, if all generated waypoints are kept on the next stage of Isochrone, the total number of potential waypoints will grow exponentially. Overcoming the "curse of dimension" has been the motivation for researchers to improve the Isochrone method for more practical use.

2.1 *Modified Isochrone voyage optimization*

The modified Isochrone algorithm introduced the sub-sector to only choose the good waypoints on the current stage for each sub-sector ΔD (Hagiwara 1989). The principal theory of this improved Isochrone method is shown in Fig. 1. To meet the requirement of fast routing planning, ship speed is assumed constant during the voyage unless encountering harsh weather conditions. The Isochrone method can be conducted as follows:

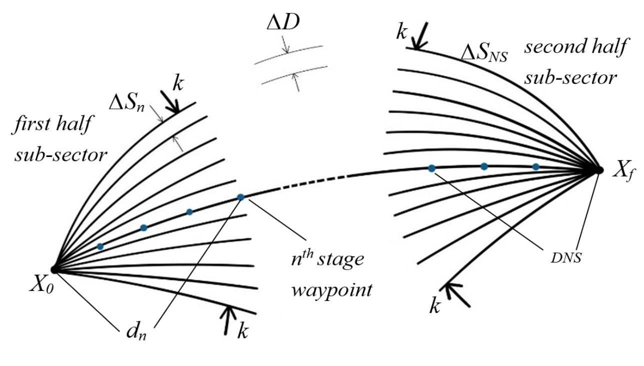

Figure 1. Definition of sub-sectors along the voyage.

(1) Determine the start point X_0 and destination X_f as illustrated in Fig. 1, and plot the great cycle line as a reference route. Determine time stage interval Δt; MetOcean data encountered in every time stage could be easily extracted. It can facilitate very fast estimation during the voyage optimization process.

(2) Simulate a ship departing from the departure location $X_0=[x_0, y_0]$ at start time T_0 to the destination location $X_f=[x_f, y_f]$ at a constant speed, where x and y represent the longitude and latitude of the waypoints of the sailing area. The ship can follow headings $C_{ref} \pm j\cdot\Delta C$ ($j= 0, 1, \ldots, m$) navigating distance $\Delta t\cdot v$ in the first step, where C_{ref} is the course of reference great cycle route connecting X_0 to X_f, v is the navigation speed, and ΔC is the increment of heading. The first Isochrone is defined. The potential arrival waypoints at time $T_0 + \Delta t$ is represented by $X_1(i)$($i = 1, 2, \ldots, 2k$). The waypoint set $\{X_1(i)\}$ defines the first Isochrone. The engine power and fuel consumption for arriving at waypoints in $\{X_1(i)\}$ can be calculated and stored according to environmental data and ship performance model based on different operational and environmental conditions encountered at different waypoints associated with each time stage.

(3) Define a set of sub-sectors by plotting great circle lines from departure location X_0 to X_f, following $C_{ref} \pm s\cdot\Delta S_1$ ($s= 0, 1, \ldots, k$), as shown in Fig. 1, and the definition of ΔS_1 is given in Section 3.2.1. In each sub-sector, only the waypoint $X_1(i)$ with minimum cost is preserved as the potential waypoint for the next Isochrone. In this paper, the function of cost evaluation is designed differently in different improvement approaches. The number of potential arrival waypoints in the Isochrone can be a constant controlled by a specific parameter k instead of exponentially as the isochrone development advances.

(4) Repeating procedures (2) and (3) until the minimum geographical distance between waypoints in the current Isochrone $\{X_n(i)\}$ to the destination is larger than half the geographical distance between X_0 and X_f. Regenerate the sub-sectors following $C_{inv} \pm s\cdot\Delta S_n$ ($s= 0, 1, \ldots, k$), where C_{inv} is the back azimuth course angle of great cycle route connecting X_f to X_0. The reason for this regeneration is defining boundaries that routes generated inside do not have sharp turnings. Connect the waypoints $\{X_n(i)\}$ in the current Isochrone to the destination point X_f using the great cycle lines. The back azimuth course angle of these great cycle lines is the new reference heading C_{ni} for every waypoint $\{X_n(i)\}$ in the current Isochrone. Thus, the ship in the i^{th} waypoint of the n^{th} Isochrone can follow headings $C_{ni} \pm j\cdot\Delta C$ ($j= 0, 1, \ldots, m$). The potential arrival waypoints at time $T_0 + n*\Delta t$ is $X_n(i)$ ($i = 1, 2, \ldots, 2k$). Improvements from different methods are all made within this part of the voyage, which will be presented in Section 3.2.

(5) When the geographical distance between any waypoint in the Isochrone $\{X_n(i)\}$ and destination X_f is less than a certain number, which in this paper is chosen to be sailing time left less than 24 hours, ΔC,

the increment of heading during voyage would be decreased to 10%~50%. This is because when approaching the surrounding destination area, the width of reversed sub-sectors also narrows down significantly. Maintaining the normal increment of the heading angle will easily make successors of one node domain and others all in every sub-sector.

(6) When the geographical distance between any waypoint in the Isochrone $\{X_n(i)\}$ and destination X_f is less than $\Delta t \cdot v$, connect the waypoints in this Isochrone to destination location X_f using a great circle line.

2.2 *Parameter sensitivity in Isochrone methods*

An isochrone can be defined as a set of connected points that a ship can reach within constant sailing time, starting from one point and going in all possible headings. These isochrones are generated from each point of the previous set. The length of the line between the previous point and each point on the Isochrone depends on ship speed since sailing time is fixed. Furthermore, the shipping speed is dependent on weather factors and engine setup, etc. In this method, there will be five essential parameters to be well specified and presented in Fig.2.

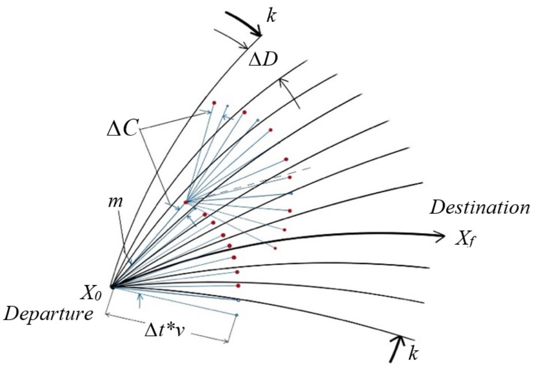

Figure 2. The graphic interpretation of Isochrone method.

They have a significant impact on voyage optimization results because they control the generation of waypoints in the search grid. The algorithm should be able to generate a certain number of candidate paths that cover potential sailing areas. The following parameters are used to define search area and waypoints by the Isochrone method,

- ΔC: the increment of heading angles between two adjacent sub-path from each of the current "optimal" waypoints at each time stage. It can influence the search area ahead of the current waypoint. For example, if it is set to larger, generated searching waypoint in the next time stage will expand much faster in width and reach the width limit sooner. In addition, generated waypoints can easily spread among more sub-sections. Too large values of ΔC may lead to locally optimized predecessor waypoints. Many other candidate waypoints in nearby sub-sectors may be ruled out, leading to partly repeatable paths.
- m: number of successor waypoints for each of the current waypoints in the current stage, corresponding to the number of spreading headings. It influences the expanding speed of searching waypoints in earlier time stages and helps to generate more feasible candidate paths by using a larger value of m. But if m becomes too large, the searching performance might be significantly reduced because a great computation effort is needed to search big areas/waypoints.
- Δt: traveling time between two adjacent time stages. It controls the looseness of the searching grid along the direction toward the destination. Large Δt may lead to slow converging of optimal searching towards the destination, and avoid too many overlapped candidate routes. However, it may cause sharp path turning in the candidate routes.
- ΔD: width of the searching limit within each local sub-sector, as shown in Fig. 2. The width limit defines the searching grid along the voyage. Same as Δt, small ΔD defines a narrow searching range perpendicular to the reference route. Small ΔD may cause overlapped sub-paths, while large ΔD leads to a sharp turning of the candidate route toward the destination.
- k: number of sub-sectors. Since each sub-sector preserves one optimal waypoint for each time stage, increasing k means that the number of waypoints in each stage of Isochrone will increase too. Similar to m, large k may improve performance and may also cause sharp turning and increase computational effort during the optimization process.

All those parameters should be chosen within a reasonable range to allow for efficient voyage optimization. They also depend on the length of the voyage and weather dynamics. Increasing the values of those parameters may have a positive impact on optimization results, but a trade-off between performance and computational efficiency should be defined.

In the original Isochrone method, a reference path is chosen for the discretization, and the reference path can be either the shortest route (the great circle route) or a typical sailing route. The optimization algorithm searches the next sub-routes with a series of headings $C \pm j \cdot \Delta C$ around the reference route as in Fig.2. After setting the basic configuration to discretize a ship's sailing area in both space and time, some searching criteria should be defined to limit the number of candidate waypoints to proceed within the voyage optimization process. Otherwise, the number of possible waypoints (to form sub-routes) will increase exponentially as the evolution of time stages. To reduce the point number, Hagiwara (1989) developed k parallel lines of equal spacing ΔD on both sides of the reference ship route to form $2k$ sub-sectors. A ship is supposed to move within the area with a width of $2k \times \Delta D$. At each time stage, only the waypoint closest in the distance to the destination is selected on each sub-sector to compose the next Isochrone/time stage.

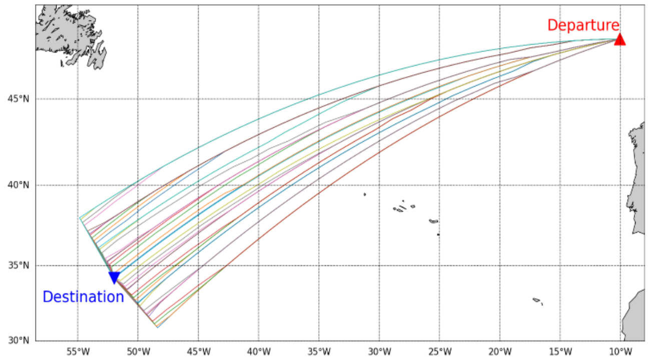

Figure 3. Potential routes using original Isochrone method proposed by Hagiwara (1989).

The modified Isochrone method (Hagiwara 1989) has apparent shortcomings for ship voyage optimization. For example, sub-routes in the last time stage have a very wide searching range, as seen in Fig. 3. Most of those candidate routes with sharp turning in are not realistic for practical voyage planning. Several strategies to improve the original Isochrone algorithms are investigated in this study and presented in the following sections.

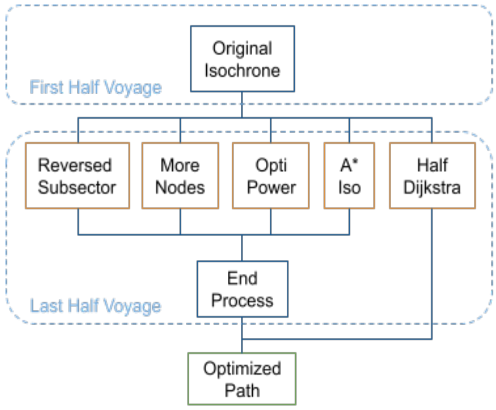

Figure 4. Concepts for different improvement strategies of Isochrone method for voyage optimization.

3.1 *Reversed subsector*

In original Isochrone by Hagiwara (1989), the sub-sector is defined as:

$$\Delta S_n = c\Delta D/\sin(cd_n),\ \text{c}=\pi/(60*180), \tag{1}$$

where d_n is the expected travel distance in n^{th} stages of isochrone waypoints set $\{X_n(i)\ (i = 1, 2, . . ., 2k)\}$ after $n*\Delta t$ hours:

Table 1. Methods used in different stages of a voyage in strategies

Name of improvement	First half voyage	Second half voyage		Final Stage
		Sub-sector	**Searching method**	
Reversed Sub-Sec	Original Isochrone	Reversed Sub-Sector	Original Isochrone	$\Delta C = \Delta C$ *10%~50%
More Nodes			Reserve more nodes	
Opti Power			Optimal power greedy search	
A* Isochrone			Additional heuristic function	
Half Dijkstra		Dijkstra		

3 STRATEGIES FOR IMPROVEMENT

Two methods to improve the original Isochrone method are investigated in this study, i.e., by modifying the generation of sub-sectors, and by better defining the optimization criteria and cost function to choose optimal waypoints in each individual sub-sector. A flowchart for different approaches to improvement of Isochrone method studied in this paper is presented in Fig. 4. For the improvement of the modified Isochrone method, the optimization process is divided into two parts from the departure to the destination. In the first half voyage, the original Isochrone method, as in Hagiwara (1989), is implemented. Then five strategies are used to optimize the results for the second half of the voyage, and their concepts are presented in Fig.4. The improvement strategies and used methods are briefly summarized in Table 1, which will be presented in detail in the following subsections.

$$d_n = n * \Delta t * V_s, \tag{2}$$

in which V_s is the ship service speed fixed during the voyage optimization; ΔD is the local sub-sector width (i.e., resolution of the Isochrone).

Note that ΔS_n represents the heading angle range in the n^{th} stage for each sub-sector, while ΔD, the parameter predefined as the width of the local sub-sector, indicates the width limit for each sub-sector in the distance. This equation indicates that the width ΔS_n is a monotonically increasing function of traveled distance d_n.

Therefore, in the last half of the voyage, if replacing traveled distance d_n with distance from the destination (*DNS*) to calculate the width and plot from destination X_f, a reversed and symmetric sub-sector set can be generated, which gradually shrinks its range when approaching the end:

$$\Delta S_{ns} = \frac{c\Delta D}{sin(cd_{ns})}, d_{ns} = d_{total} - d_n, \tag{3}$$

where d_{total} is the travel distance from X_0 to X_f. In this method, the cost function is chosen to be the shortest distance to the destination.

3.2 *More nodes preserved*

One problem caused by the reversed sub-sector is, as the sub-sector width shrinks towards the destination, some nodes in the current stage with the lowest cost will quickly rule out other candidates and domain all the nearby sub-sectors; therefore, paths later developed will all derive from its successors. In the resulting path set, too many paths are overlapped, especially in the first half of the voyage, which can be considered as some early stages are trapped in a local minimization. This problem was addressed by the following procedures:
(1) The number of nodes/waypoints preserved in each sub-sector is increased and controlled by an extra parameter to prevent the waypoints from growing exponentially.
(2) Every predecessor node is only allowed to keep a limited number of its successors, avoiding its domination.
(3) This gives other candidate nodes chances to survive for a longer time and, to some extent, prevent being denied because of temporary suboptimality.

3.3 *Optimal power search*

Another way to generate more candidate paths is, in the latter half of the voyage, node searching is replaced by a greedy algorithm: remove the restriction of sub-sector, every waypoint expands following the heading $C_{ni} \pm j \cdot \Delta C$ $(j = 0, 1, \ldots m)$, keeps the node with lowest fuel consumption and proceeds until reaching the destination.

3.4 *A* isochrone*

Alternatively, A* algorithm can be used to optimize the voyage's second half. A* is a graph-searching algorithm used widely in many fields which has a competitive performance in optimality and computational efficiency. It is an informed search algorithm that establishes an evaluation function that involves both forward and backward cost estimation along the path:

$$f(n) = g(n) + h(n), \tag{4}$$

where $g(n)$ is the cost accumulated from departure, and $h(n)$ is the heuristic function which is problem-specific and estimates the cost from the current node to the destination.

In the original isochrone method, the cost function used in evaluating a node's cost is the fuel consumption from departure or current distance to the destination. In this approach, referring to A*, the cost function is extended with a heuristic term $h(n)$, and $f(n)$ accordingly becomes the overall fuel consumption for the whole voyage, $g(n)$ is the fuel consumption from departure, and $h(n)$ is the fuel consumption needed to reach the destination by estimation through ship performance model, basing on the weather forecast.

3.5 *Half Dijkstra isochrone*

In the original isochrone algorithm, all nodes are connected with a directional grid, which means the edges between nodes are a one-way route, pointing from the previous generation to the next. If a node is removed from the grid, all its predecessors will be automatically deleted since they will no longer form a complete path from departure to destination. This is one of the reasons the resulting path set easily contains overlapped paths.

Dijkstra is the graph searching algorithm where A* extends from, which is also widely used because of its efficiency and little complexity. Different from Isochrone, its network is undirected, and there is no binding between nodes. Therefore, in this approach, to avoid too many overlapped results, in the latter half of the voyage, a grid is developed based on the middle stage isochrone waypoints, as illustrated in Fig. 5 below:

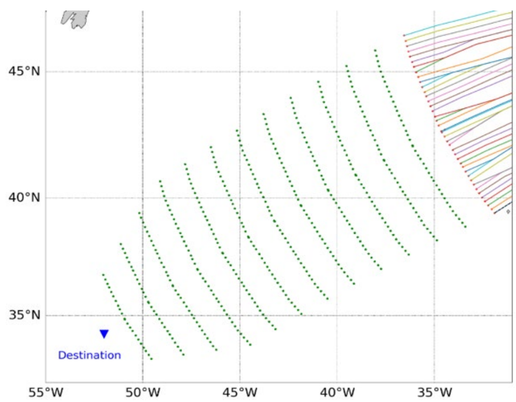

Figure 5. Node grid generated in the latter half of voyage to use Dijkstra algorithm.

After generating this grid, assign the weight of the edge based on the fuel consumption estimation through the ship performance model according to weather conditions and traveling time. Search for a path to destination starting from each waypoint in the middle Isochrone using Dijkstra algorithm, with a cost function evaluating fuel consumption from departure to current position. These candidate paths obtained, however, would possess a different time of arrival since the binding between nodes is removed, and the length of edges between nodes, therefore, varies while traveling speed most of the time remains constant. In this approach, the optimal path then becomes the one with the closest time of arrival.

4 COMPARISON OF VARIOUS IMPROVED ISOCHRONE OPTIMIZATION METHODS

4.1 *Case study of ship voyages*

A chemical tanker with full-scale measurement of 2 voyages is used as a case study to compare the capability of voyage planning by the improved Isochrone optimization methods. The case study voyages are chosen for the ship with main particulars, as in Table 2, sailing in North Atlantic. A conventional weather routing system was installed on the ship to provide guidance on its sail planning. Combined with the ship master's experience, the actual sailing routes are supposed to be more efficient than ordinary voyage planning systems. The case study voyages were measured in winter 2016, shown in Fig. 6. For ship voyage optimization, the sailing time and fuel cost along candidate sub-paths should be estimated to select optimal waypoints. The estimation requires inputs of all encountered MetOcean environments (wind, wave, and current) around the searching area and the ship performance model that describes the ship's speed and fuel consumption relationship in terms of encountered MetOcean conditions. In this study, all related MetOcean parameters used in the case study are extracted from ECMWF ERA-5 (2019) dataset, such as wind (wind speeds and directions) and wave information (wave height and period), and from http://marine.copernicus.eu/ (Copernicus 2019) server for current information. Furthermore, the ship energy performance model developed by *Lang et al. (2020)* is used for voyage optimization.

Table 2. Principal particulars of the chemical tanker ship

Length L_{oa}	178.4 m	Design draught	10.98 m
Length L_{pp}	174.8 m	Block coefficient	0.8005
Beam B	32.2 m	Deadweight	50752 t
Depth	17.0 m		

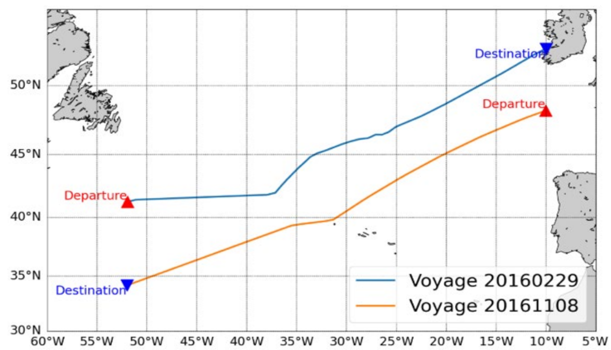

Figure 6. Case study voyages for comparison.

Results of case studies are based on the whole voyage, since Isochrone algorithm generates waypoints in steps, and nodes in each step are bound like a chain. The second half voyage optimization is not independent of the first half; therefore, they should be presented together as a complete route.

4.2 *Results of optimization for a westbound voyage*

A westbound case, Voyage 20161108, planned by a weather routing system from the shipping market, as well as original isochrone voyage optimization by Hagiwara (1989) are studied. Since the optimization result is sensitive to the parameter settings of isochrone methods, for Voyage20161108 case study, the value of each parameter is listed in Table 3, and the details of each optimized voyage obtained by different approaches are presented in Fig. 7 and Fig. 8. In Table 4, for each voyage, estimated time arrival, fuel consumption and travel distance in total are listed accordingly.

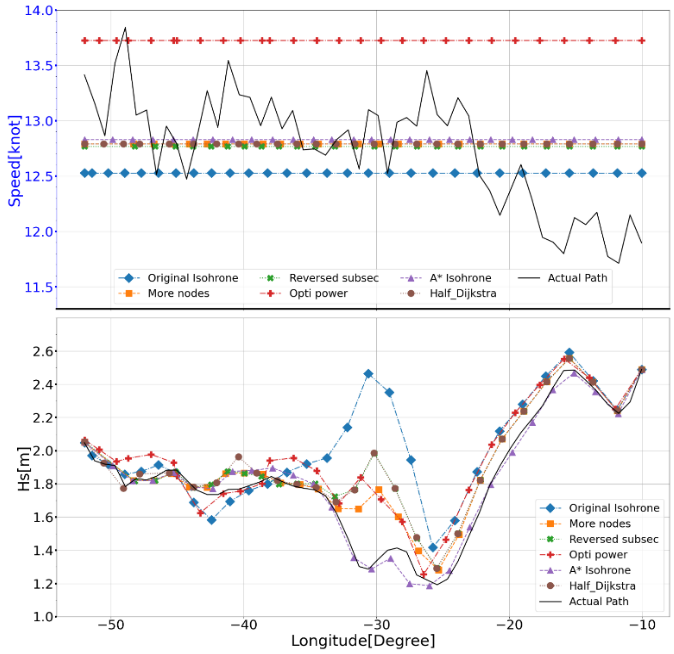

Figure 7. Sailing speeds and encountered H_s along the routes

It can be seen from Fig. 7 that the weather condition Voyage20161108 encountered is relatively calm, and H_s (significant wave height) is lower than 3 meters along the voyage. The optimal voyage generated by reversed sub-sector, reserving more nodes and half Dijkstra are similar in most of the part, only deviating from each other around the end stage; therefore, their traveling distance are close as well, from Table 4. It indicates that reserving more candidate nodes in each stage does not really influence the result since the node number is the only changing factor compared with reversed sub-sector. The reason why half Dijkstra is not improving could be that the first and second half planning is separated from each other as well as the cost evaluation function; generally, this method is initialized and restarted again in the middle stage isochrone. Therefore, trying to combine these two parts could be a possible consideration for further improvement.

The optimal power searching approach, however, shows the highest fuel consumption. The greedy algorithm it refers based on the assumption that the optimal solution obtained by all sub-tasks it divides into can lead to a global optimization in the end. However, in the isochrone algorithm, from the result, this assumption may be proven not to stand.

Table 3. Parameter of isochrone algorithm

ΔC [*degrees*]	*m*	$\Delta t[h]$	ΔD	*k*
2	20	6	7	20

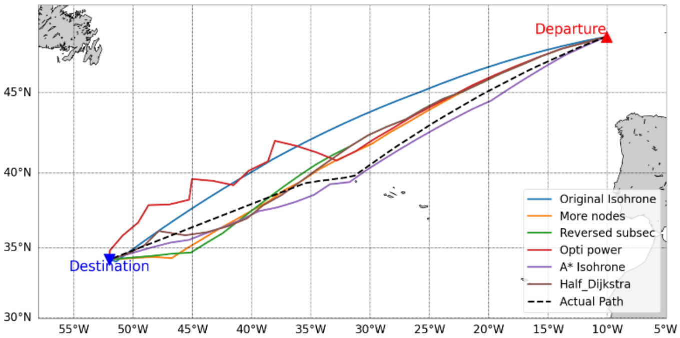

Figure 8. The optimized voyage from different approaches.

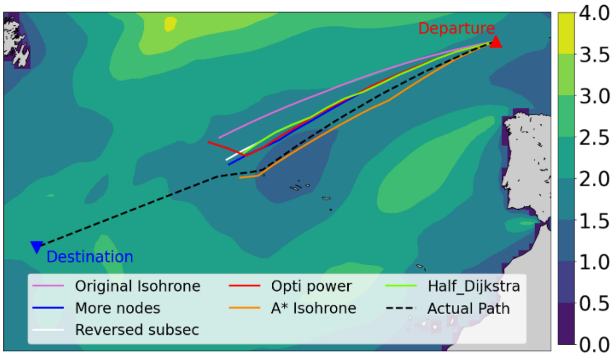

Figure 9. Optimized voyage with the weather condition in the middle stage of the voyage.

Table 4. Results of the proposed voyage optimization algorithm

Category	ETA[h]	Fuel [ton]	Distance [km]
Actual route	164.25	117.722	3877.45
Original Isochrone	164.30	118.641	3812.11
Reversed Sub-Sec	164.77	122.715	3895.98
More Node	164.39	120.307	3894.16
Opti Power	164.66	144.130	4185.09
A* Isochrone	164.54	116.110	3909.47
Half Dijkstra	164.74	118.570	3902.37

A* isochrone, on the other hand, in this case, shows the best performance. From the route, it is also the one close to the actual voyage. They all head south toward the destination instead of directly following the great circle route, and the reason would be the weather condition which can be proved by MetOcean data. As the assumption of Isochrone, the sailing speed remains constant unless an extreme condition is encountered; therefore, the speed of all methods is similar to straight lines.

The main difference appears in the area at around -30°W longitude, the middle stage of the voyage, and is mainly caused by H_s. The weather condition at this time is presented in Fig. 9, which gives the reason this approach stands out from others. The voyage A* isochrone generated meets the lowest H_s in the middle, and it is the only one sails the same direction as the actual route, which is operated manually. Besides this, during the whole voyage, it encounters a relatively steadier sea state. This shows that A* isochrone has the capability to optimize voyage considering dynamic weather conditions, with the objective of energy efficiency.

4.3 *Results of optimization for an eastbound voyage*

The optimization results for an eastbound case, Voyage20160229, are presented in this section. In this case, values of each parameter keep the same as the westbound case, listed in Table 3, and details of optimized voyages obtained by different approaches are presented in Fig. 10 and Fig. 12. Weather conditions at the time when voyages behave most differently are presented in Fig. 11. Estimated time arrival, fuel consumption, and travel distance are listed in Table 5.

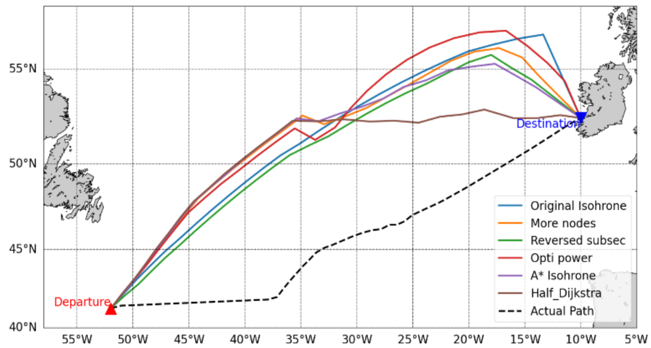

Figure 10. The optimized voyage from different approaches

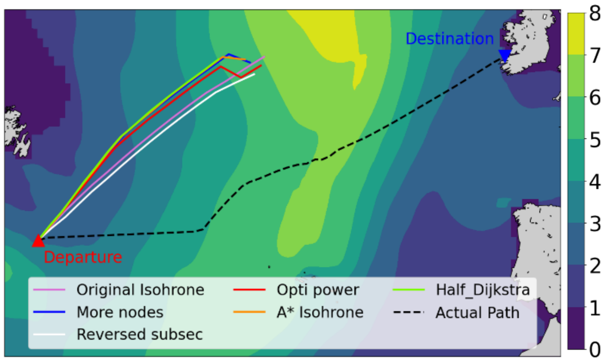

Fig. 11, Optimized voyage with the weather condition in the middle stage of the voyage.

In this eastbound case, the encountered weather conditions were rougher. From Fig. 12, H_s could even reach 10 meters during the actual voyage. Optimal paths generated all head north to avoid extreme weather and then turn around. It is noticed that all generated voyage deviates from the actual route, and the reason could be the actual route is sailing much faster at the beginning, and the speed is not constant; thus, its weather condition is different all along, which consequently leads to different choices.

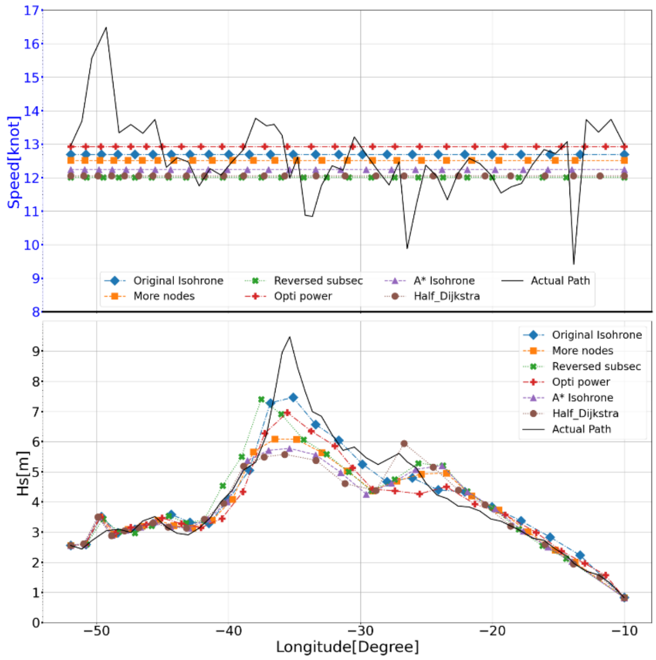

Figure 12. Sailing speeds and encountered H_s along the routes

The optimal voyage from original Isochrone leads to a sharp turning. As results in Table 5, the optimal power strategy generates a route winding around the destination and therefore has a higher fuel consumption. The performance of keeping more candidate nodes is similar to reversed subsector and slightly improves energy efficiency. A* isochrone gives the lowest fuel consumption voyage among the rest with a smoother turning. Meanwhile, the voyage from half Dijkstra deviates from others in the middle of the voyage and gives a lower fuel consumption, which might indicate this method has the capability to explore voyages differently than isochrone types.

For this case, all strategies can improve Isochrone method to not only generate a smoother route but also meet the objective of energy efficiency and accurate ETA. A* isochrone has the best performance among them, saving approximately 14.4% fuel compared to the actual voyage case.

Table 5. Results of the proposed voyage optimization algorithm

Category	ETA[h]	Fuel [ton]	Distance [km]
Actual route	159.00	173.719	3624.91
Original Isochrone	159.84	175.522	3757.49
Reversed Sub-Sec	160.44	161.509	3569.50
More Node	158.90	155.883	3684.01
Opti Power	159.66	166.157	3822.34
A* Isochrone	158.02	147.546	3583.96
Half Dijkstra	157.17	148.024	3507.99

5 CONCLUSIONS

Different strategies to improve the original Isochrone method for voyage optimization have been investigated to design a ship's sailing route in terms of various objectives, such as minimum fuel consumption and accurate ETA. Five parameters significantly impact the voyage optimization results by the Isochrone methods. Different strategies to improve the Isochrone algorithm are proposed, and their efficiency is compared based on two case study voyages. The parameters within each improved algorithm are kept the same to make a comparison. It should be noted that those parameters should be adjusted for different improvement strategies, to achieve the best optimization results. It requires further study with possible quantified and formulated instructions.

For the two case study voyages used for comparison, optimal power searching is not able to provide a globally optimized solution, and reserving more nodes is shown with slight improvement, together with using reversed sub-sector. Half Dijkstra behaves differently from other methods, and potential improvement could be considered in better combining the Isochrone and Dijkstra used in separate parts. A* isochrone optimization algorithm gives more reliable optimal voyage planning results. Moreover, to keep the efficiency of Isochrone algorithms, the computational cost of all purposed strategies does not increase significantly, and remains the same level as original algorithm, which is within minutes. However, there is still potential to further improve it after the strategy with the best performance is determined. Also, it can be further investigated with more voyage cases in various weather conditions, while as mentioned above, giving possible parameter adjustment instructions.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the funding from the project AUTOBarge, European Union's EU Framework Program for Research and Innovation Horizon 2020 under Grant Agreement No. 955768.

REFERENCES

Bellman, R. (1952). On the theory of dynamic programming. Proceedings of the national Academy of Sciences, 38(8), 716-719.
Calvert, S., Deakins, E., & Motte, R. (1991). A dynamic system for fuel optimization trans-ocean. The Journal of Navigation, 44(2), 233-265.
Chen, H. (1978). A dynamic program for minimum cost ship routing under uncertainty (Doctoral dissertation, Massachusetts Institute of Technology).
De Wit, C. (1990). Proposal for low cost ocean weather routing. The Journal of Navigation, 43(3), 428-439.
Dijkstra, E.W. (1959). "A note on two problems in connexion with graphs," Numerische Mathematik, 1, 269–271.

Hagiwara, H. (1989). "Weather routing of (sail-assisted) motor vessels," Ph.D. dissertation, Delft University of Technology, Delft, the Netherlands.
James, R. W. (1957). “Application of wave forecasts to marine navigation,” US Navy Hydrographic Office, Washington, USA.
Klompstra, M. B., G. Olsder and P. Van Brunschot (1992). “The isopone method in optimal control,” Dynamics and Control, 2(3), 281-301.
Lang, X., Wang, H., Mao, W. and Osawa, N. (2020). “Impact of ship operations aided by voyage optimization on a ship’s fatigue assessment,” Journal of Marine Science and Technology, DOI:10.1007/s00773-020-00769-8.
Larsson, E., M. H. Simonsen and W. Mao (2015). “DIRECT optimization algorithm in weather routing of ships,” Proceeding of the Twenty-fifth International Ocean and Polar Engineering Conference, International Society of Offshore and Polar Engineers, USA.
Marie, S. and E. Courteille (2009). “Multi-objective optimization of motor vessel route,” Marine Navigation and Safety of Sea Transportation, 9, 411-418.
Priftis, A., Papanikolaou, A. and Plessas, T. (2017). “Parametric Design and Multiobjective Optimization of Containerships,” Journal of Ship Production and Design, 33(1), 46-59.
Rabitz, H. (1989). “Systems Analysis at the Molecular Scale,” Science, 246(4927), 221-226.
Shao, W. and Zhou, P. (2011). “Development of a 3D Dynamic Programming Method for Weather Routing,” Methods and Algorithms in Navigation: Marine Navigation and Safety of Sea Transportation, 181.
Wang, H., Lang, X. and Mao, W. (2021). “Voyage optimization combining genetic algorithm and dynamic programming for fuel/emissions reduction,” Transportation Research Part D: Transport and Environment, 90, No.102670.
Wang, H., Mao, W. and Eriksson, L. (2019). "A Three-Dimensional Dijkstra's algorithm for multi-objective ship voyage optimization," Ocean Engineering, 186, No. 106131.