HIERARCHICAL ROUTING ON ADAPTIVE QUADTREE GRAPHS FOR EFFICIENT LARGE-SCALE PATHFINDING

Kelvin Lee, Ph.D.
Institute of High Performance Computing (IHPC)
Agency for Science, Technology and Research (A*STAR), Singapore
Email: kelvin_lee@ihpc.a-star.edu.sg

Muhammad Adam Bin Marih
Singapore Institute of Technology, Singapore
Email: 2202278@sit.singaporetech.edu.sg

Ke Wang, Ph.D.
Institute of High Performance Computing (IHPC)
Agency for Science, Technology and Research (A*STAR), Singapore
Email: wang_ke@ihpc.a-star.edu.sg

Xiuju Fu, Ph.D., Corresponding Author
Institute of High Performance Computing (IHPC)
Agency for Science, Technology and Research (A*STAR), Singapore
fuxj@ihpc.a-star.edu.sg

Xiaorong Li, Ph.D.
Infocomm Technology Cluster
Singapore Institute of Technology
xiaorong.li@singaporetech.edu.sg

Zheng Qin, Ph.D.
Institute of High Performance Computing (IHPC)
Agency for Science, Technology and Research (A*STAR), Singapore
qinz@ihpc.a-star.edu.sg

Word Count: 4145 words + 3 table(s) × 250 = 4895 words

Submission Date: July 28, 2025

ABSTRACT

Efficient pathfinding in large and complex spatial networks is a fundamental challenge across many domains, including urban transportation and maritime navigation. To address this, we develop a hierarchical routing algorithm based on adaptive quadtree grids, with a focus on long-distance marine vessel routing as a case study. This task typically involves computing shortest paths across discretized representations of the Earth's surface. While uniform-resolution grids are commonly used, they produce extremely large graphs that incur high computational and memory costs. We propose an adaptive quadtree-based grid structure with higher resolution near coastlines and narrow passages while remaining coarse in open-ocean regions to significantly reduce the total number of nodes and edges. We detail the construction of a quadtree grid to represent navigable waters, define graph connectivity across variable-resolution cells, and apply standard shortest-path methods such as A*. To further improve efficiency, we incorporate hierarchical routing on the quadtree grid, enabling fast search in a two-stage process. Experimental evaluations on representative global routes demonstrate that the quadtree approach reduces graph size by several orders of magnitude, lowers computation time substantially, and preserves route accuracy. Additional efficiency gains are achieved through hierarchical routing on the quadtree grid. The proposed method provides a scalable and effective foundation for global pathfinding applications and can be extended to incorporate dynamic, environment-dependent travel costs in future work.

Keywords: Quadtree; Adaptive spatial discretization; Graph-based routing; Hierarchical routing; Maritime transportation

INTRODUCTION

Efficient routing of ocean-going vessels is a critical component in global maritime operations, as shipping accounts for over 80 percent of international trade by volume (1), and is estimated to emit about 10% of the global emissions in 2050 (2). Coupled with the growing regulatory pressure from frameworks such as the IMO's Carbon Intensity Indicator (CII, (3)) and the EU Emissions Trading System (ETS, (4)), companies are under growing obligations to improve energy efficiency and reduce greenhouse gas emissions. With developments in alternative fuels and emerging energy-saving technologies like wind-assisted propulsion and air lubrication on the horizon, efficient routing provides shipping companies with an immediate and practical way to achieve operational cost savings and reduce environmental impact.

Achieving efficient routing starts with computing high-quality vessel routes at a global scale. This foundational step involves identifying shortest or least-cost paths using reliable pathfinding algorithms that account for navigational constraints across complex geographies. Building on this foundation, dynamic environmental factors such as weather and currents can be integrated to further enhance route optimization. As a result, efficient and scalable routing algorithms are essential for enabling real-time voyage planning, fleet optimization, and long-term emissions reduction strategies.

To this end, we propose an adaptive quadtree-based grid that adjusts spatial resolution based on geographic complexity. Fine cells are used near coastlines and narrow passages, while coarser cells cover homogeneous open-ocean areas. This approach significantly reduces graph size without compromising routing accuracy. We describe how quadtree grids are generated, how graph connectivity is defined between variable-resolution cells, and how shortest-path computations are performed on these grids using classical methods such as A*. To further enhance efficiency, we integrate hierarchical search techniques specifically adapted to the quadtree structure.

Through experiments on representative global routes, we show that our method substantially improves computation time and memory efficiency compared to uniform grids, while maintaining high path quality. The proposed framework offers a scalable and practical solution for global vessel routing and provides a strong foundation for future extensions that incorporate dynamic environmental factors such as wind and currents.

BACKGROUND AND RELATED WORK

Vessel routing as a pathfinding problem on grids

Shortest-path algorithms are widely used in marine navigation to determine optimal routes between ports or waypoints on the Earth's surface (5–7). To apply these algorithms, the continuous ocean domain is typically discretized into equal-sized cells, where each cell corresponds to a graph vertex and edges represent navigable paths. Edge costs may encode distance, estimated travel time, fuel consumption, or environmental factors depending on the objective. In the case of distance-based routing, edge weights correspond to geodesic distances, and the problem reduces to finding the minimum-cost path through the graph.

Uniform grids and their limitations

A common approach is to use uniform-resolution grids, where the entire ocean surface is divided into cells of equal size. Uniform grids are straightforward to generate and ensure consistent connectivity patterns, making them easy to implement in routing algorithms such as Dijkstra's algorithm or A*. However, achieving high spatial resolution across a global domain leads to extremely large graphs. This increases memory requirements and computation times significantly, even though such fine resolution may only be necessary in limited areas, such as coastal zones or constrained waterways. As a result, uniform grids are inefficient for global-scale routing tasks that need both fine detail and scalable performance.

Adaptive spatial discretization approaches

An alternative strategy is to reduce graph size by adapting the spatial resolution of the grid itself. Adaptive grids refine the discretization selectively, allocating finer resolution in areas of high routing complexity (e.g., near coasts or narrow straits) and coarser resolution in homogeneous open-ocean regions where fine detail is unnecessary. The quadtree data structure is a well-established for spatial encoding and compression, supporting efficient storage and query of variable-resolution grids (*8*).

In this work, we investigate the use of adaptive quadtree-based grids for global marine vessel routing, aiming to reduce graph size and computation time while maintaining high accuracy in computed routes. This is similar to previous studies (*9, 10*) that employed quadtree data structures to accelerate pathfinding by reducing spatial complexity. However, these works are typically limited to small-scale demonstrations, often evaluating performance on only one or two short-sea or localized routes. In contrast, our study provides a more systematic analysis of the benefits of adaptive quadtree grids at various resolutions through comprehensive experiments across diverse global routes.

Hierarchical routing techniques

To address computational challenges in large graphs, hierarchical routing methods have been developed (*11*). These techniques preprocess the graph to construct high-level abstractions that allow long-distance routing queries to be solved efficiently by traversing higher-level "shortcut" edges. Though, hierarchical methods alone do not reduce the total number of nodes or edges in the underlying graph.

In summary, while prior work has explored both adaptive spatial discretization and hierarchical routing techniques, these approaches have largely been treated separately and evaluated on limited scales. Quadtree-based methods in particular are often demonstrated on only a handful of routes and lack systematic evaluation across varied global scenarios. At the same time, hierarchical routing techniques have typically been developed in the context of uniform grids and do not directly address the structural challenges introduced by variable-sized grids. In this work, we bring these ideas together by integrating adaptive quadtree grids with hierarchical routing into a unified framework for large-scale voyage optimisation. Through comprehensive experiments on diverse global routes, we demonstrate that this combined approach achieves substantial reductions in graph size and query time, while maintaining high path quality, making it a scalable and practical solution for real-world maritime routing systems.

METHODOLOGY

In this section, we describe the process of generating the underlying graph on which pathfinding algorithms operate to determine the shortest route between any given origin and destination coordinates. We then explain how standard pathfinding algorithms, along with their hierarchical variants, are applied to perform routing on both uniform grids and adaptive quadtree grids.

Unlike vehicles on road networks, which are constrained to physically defined lanes and intersections, or airplanes, which follow designated airways and waypoints in controlled airspace, ships at sea enjoy much greater navigational freedom. In the open ocean, vessels can generally choose their own routes without being limited to fixed paths. Granted, ships must still comply with navigational regulations like the International Regulations for Preventing Collisions at Sea (COLREGs) (*1*), which set out rules for safe navigation and avoiding collisions. In certain areas such as those with traffic separation schemes, narrow channels, or busy ports, vessel movements are subject to more stringent regulation, with recommended or mandatory routing measures designed to manage maritime traffic and enhance safety. Beyond these localized constraints, vessels are generally free to navigate as they choose, resulting in a vast area of open, navigable waters.

Pathfinding on uniform grids

Pathfinding algorithms operate on directed graphs. To use pathfinding algorithms to generate viable vessel routes, we will require an accurate representation of the navigable waters. One common approach to constructing such a graph is to overlay a uniform grid on the map and use the centroids of these grid cells as vertices of the graph. Once it is identified whether a grid cell covers land or sea, non-navigable cells can be pruned, thereby reducing the overall graph size and improving computational efficiency.

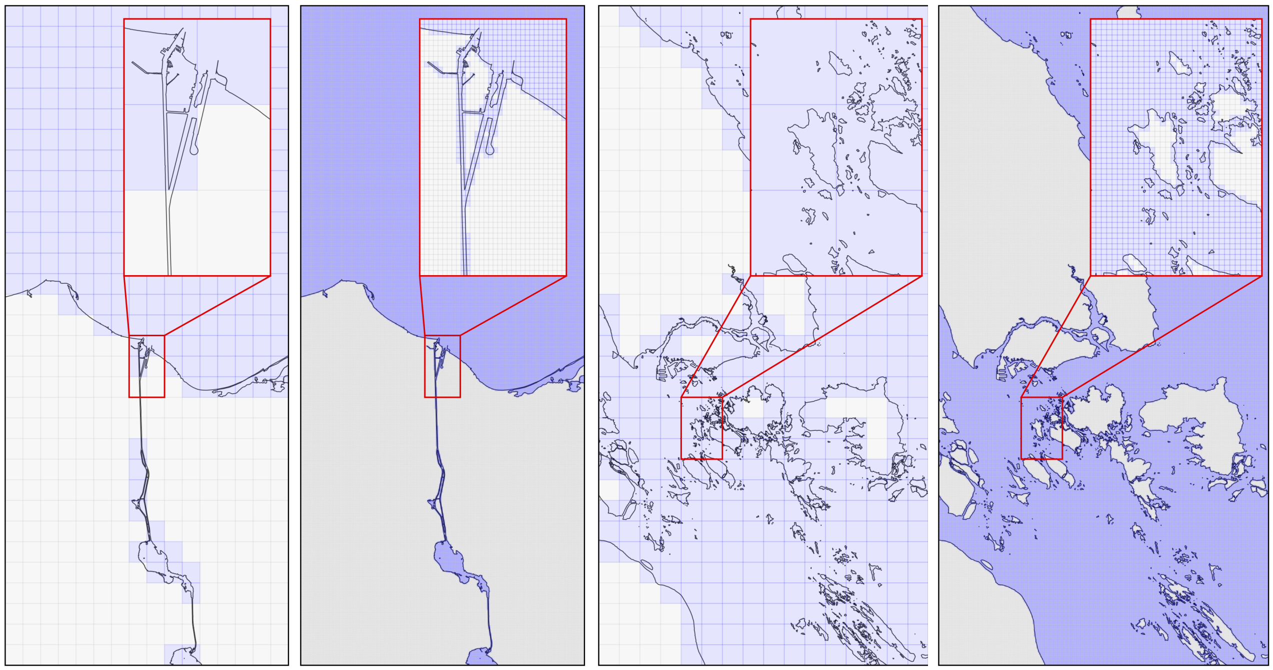
To ensure that routing algorithms function correctly across all navigable waters, it is essential that every navigable area is covered by at least one sea grid cell and that these cells maintain connectivity with their adjacent sea grid cells. This connectivity is particularly crucial in narrow waterways. For example, the Suez Canal (a critical maritime corridor handling substantial global cargo traffic annually) is only a few hundred meters wide at its widest point. If the grid cells used exceed this width, the canal may not be accurately represented, which renders routes along the Suez Canal infeasible.

Conversely, using excessively fine grids to capture narrow waterways precisely leads to very large graphs, imposing significant computational costs during route planning. Therefore, there is a fundamental trade-off between grid resolution and computational efficiency that must be carefully balanced to enable accurate yet tractable vessel routing.

Pathfinding on variable-sized grids

As can be seen from Figure 1, fine grids are only necessary at narrow waterways or around complex terrains. Using the same grid resolution required to represent these areas to represent the open seas would result in unnecessary complexity when performing pathfinding. One way to overcome this is by the use of variable-sized grids. Here, we used a quadtree-based grid to represent the entire map, which allows for coarse grids in the open sea and fine grids where necessary. A quadtree is a hierarchical spatial data structure that recursively subdivides a two-dimensional space into four quadrants or regions. It is typically used to efficiently represent areas of varying detail by allocating higher resolution only where it is needed. In our case, the quadtree enables adaptive refinement of the grid: regions with complex coastlines or narrow channels are subdivided into smaller cells to capture navigational constraints accurately, while vast open waters remain represented with large, coarse cells, minimizing memory usage and computational cost during pathfinding. This approach balances accuracy and efficiency, making it well-suited for large-scale maritime navigation scenarios.

In constructing our quadtree-based grids, we want to ensure that water bodies and waterways are represented with sufficient detail to enable pathfinding algorithms to generate feasible



(a) Suez Canal (b) Singapore Strait

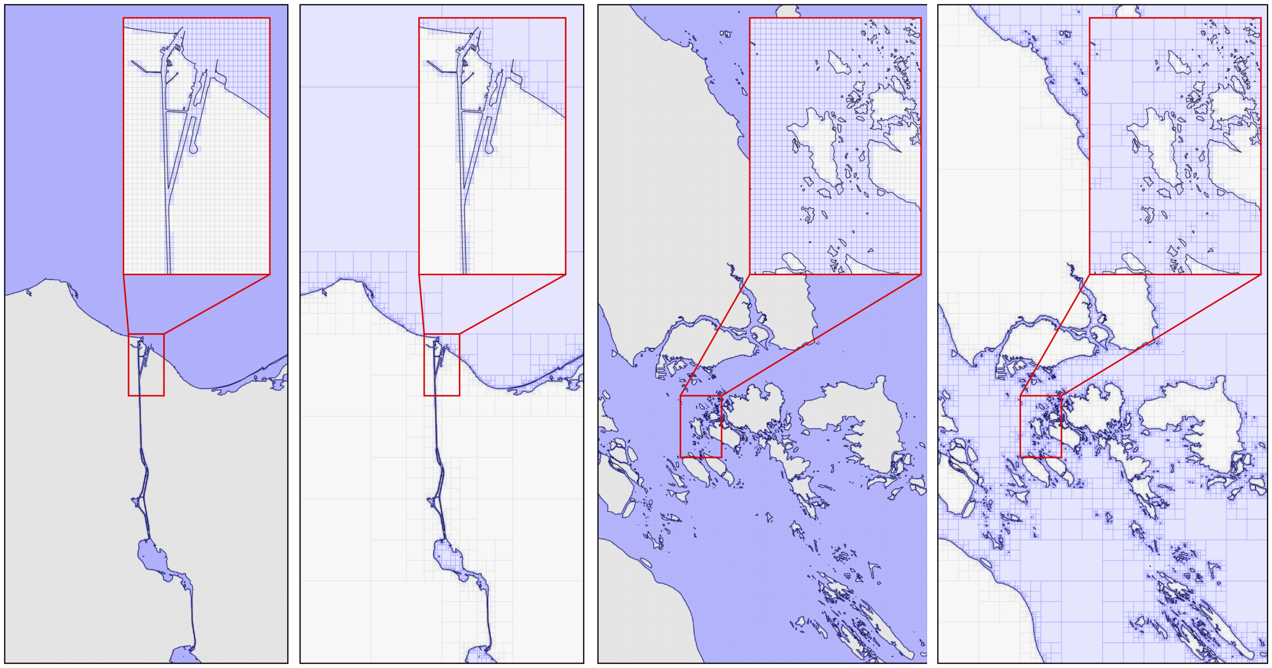
FIGURE 1: Maps of Suez Canal and Singapore Strait overlaid with uniform grids at two resolutions[1](L12 and L16). Navigable sea grid cells are shown in blue, while white grid cells denote land. With a coarser grid, navigation may be blocked through narrow channels, as seen in the Suez Canal at L12. Small land parcels may also be poorly represented, as in the Riau Islands near the Singapore Strait. Finer grids are required to capture the terrain accurately for safe and proper navigation.

routes along them. To achieve this, we begin with a single cell of size $360° \times 360°$ centred at $(0,0)$, on a high-resolution representation of global coastlines. Each cell is then recursively subdivided into four equal quadrants until one of the following stopping criteria is met:

- The grid cell reaches the maximum allowed resolution (smallest grid size);
- The percentage of land cover exceeds a fixed tolerance τ, and the area of water cover is less than a threshold θ;
- The percentage of water cover exceeds a fixed tolerance τ, and the area of land cover is less than a threshold θ.

The latter criteria are imposed to prevent significant portions of land or water (as defined by the threshold θ) contained within a larger area of different terrain from being disregarded. We find that this set of criteria yields a satisfactory quadtree-based grid. While some small landlocked or inland waterways may be misrepresented, this has no impact on routing efficacy for commercial vessels. Compared to uniform grids, quadtree-based grids have significantly fewer cells while retaining detail where necessary. From Table 1, one can observe that the number of cells in uniform grids is at least one order of magnitude more than that of variable grids. This translates to

[1]Throughout this document, different resolutions are denoted by the letter "L" followed by a number. The larger the number, the finer the grids. The size of each grid cell at level L is given by $360/2^L$ degrees.



(a) Suez Canal (b) Singapore Strait

FIGURE 2: Maps of the Suez Canal and Singapore Strait overlaid with uniform and variable-resolution grids at the same resolution (L16). The adaptive grid significantly reduces the number of cells, particularly in open-sea regions, while preserving fine detail near coastlines and complex geographic features. Navigable sea grid cells are shown in blue, while white grid cells denote land.

significantly reduced computational costs for pathfinding.

TABLE 1: Uniform and variable grids at various resolutions. **Res.**: Resolution. **GS**: Grid size in 10^{-5} degrees (for variable grids, this represents the size of the smallest grid cell for the resolution). **HGSE**: Horizontal grid size at equator in metres.[2]

Res.	**GS**	**HGSE**	**Number of cells (uniform)**		**Number of cells (variable)**	
			Land/sea	Sea only	Land/sea	Sea only
L12	8789	9784	8,388,608	5,583,981	337,388	242,550
L13	4395	4892	33,554,432	22,253,262	772,463	547,908
L14	2197	2446	134,217,728	88,811,282	1,799,375	1,264,082
L15	1099	1223	536,870,912	354,758,393	4,162,604	2,904,110
L16	549	611	2,147,483,648	1,417,879,802	9,379,538	6,513,194
L17	275	305	8,589,934,592	5,668,810,193	20,180,549	13,982,153
L18	137	153	34,359,738,368	22,668,648,838	40,719,950	27,929,620
L19	69	76	137,438,953,472	90,661,116,377	81,714,335	55,445,030

[2] The farther from the equator, the shorter the distance represented by the same grid size in degrees, because lines of longitude get closer together toward the poles.

Algorithm 1: Hierarchical Pathfinding Algorithm

Input: Start coordinates *start*, target coordinates *goal*, precomputed intra-cell distances *pds*, and routes *prs*

Output: Feasible (possibly suboptimal) path from *start* to *goal* and its corresponding travel cost

```
startCell ← GetQuadtreeCell(start) ;                         // At Level 10
goalCell ← GetQuadtreeCell(goal) ;                           // At Level 10
if startCell = goalCell then
    path ← A*(start, goal) ;                                 // Within the same cell
    return (path, path.Distance())
startPortal ← FindNearestPortal(start, startCell) ;
goalPortal ← FindNearestPortal(goal, goalCell) ;
highLevelPath ← A*(startPortal, goalPortal, pds) ;
if highLevelPath = ∅ then
    return No feasible path found
fullPath ← [] ;
distance ← 0 ;
startPath ← A*(start, startPortal) ;
fullPath.extend(startPath) ;
distance ← distance + startPath.distance() ;
foreach (portal_i, portal_{i+1}) ∈ highLevelPath do
    fullPath.extend(prs[portal_i][portal_{i+1}]) ;
    distance ← distance + pds[portal_i][portal_{i+1}] ;
endPath ← A*(goalPortal, goal) ;
fullPath.extend(endPath) ;
distance ← distance + endPath.distance() ;
return (fullPath, distance)
```

Hierarchical pathfinding

Despite significant reductions in computational cost enabled by variable-sized grids, an inherent trade-off between solution quality and algorithmic efficiency remains. To obtain high-quality routes that closely approximate the true shortest navigable path, it is necessary to operate at a very fine spatial resolution. However, doing so increases the number of grid cells, thereby increasing both memory usage and runtime during pathfinding.

One way to address this challenge is through hierarchical pathfinding, which allows the algorithm to first operate at a coarse level before progressively refining the solution at finer levels of detail. A useful analogy for understanding hierarchical pathfinding is the way humans typically navigate using paper maps. When planning a journey between two distant locations, we first identify the major roads or highways that connect the general areas, and only then work out the specific streets required to enter and exit the main route. This process naturally decomposes the problem into a global planning followed by local refinements.

The same principle can be applied in the maritime context. Long-distance voyages can

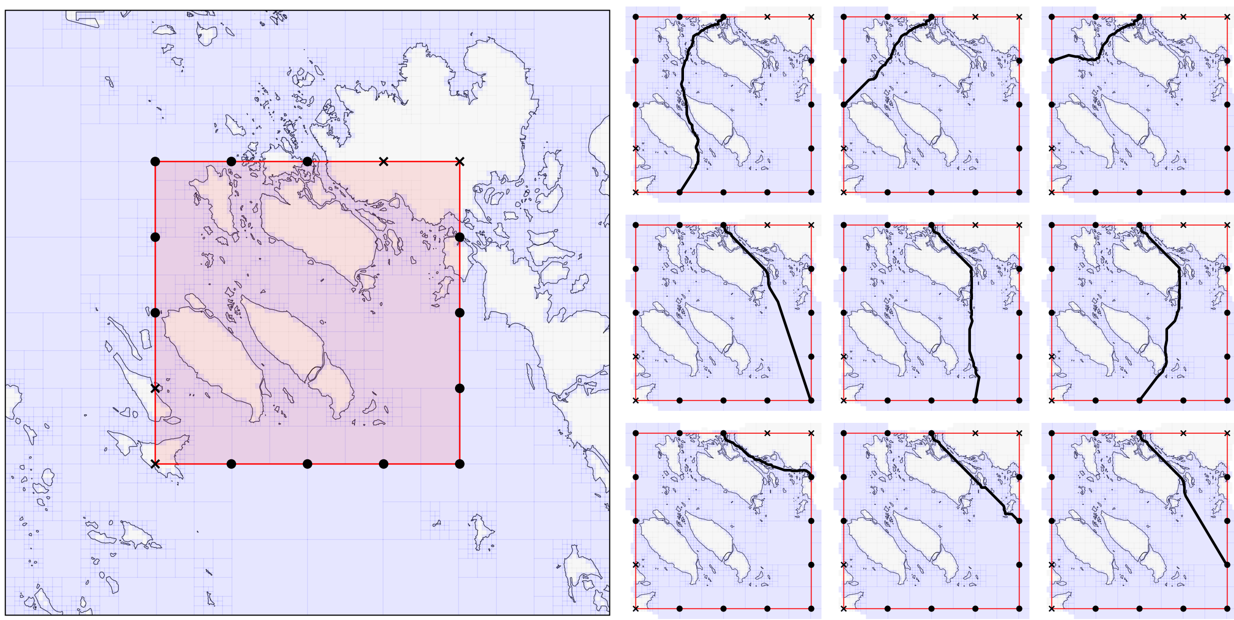


FIGURE 3: The figure on the left illustrates the portals of a grid cell overlaid on the Riau Islands near the Singapore Strait. Dots indicate portals located on navigable waters, while crosses mark those on land and are subsequently pruned. The figures on the right show the precomputed shortest paths between pairs of navigable portals. Note the significant variation in path costs due to the complex terrain contained within the cell.

be broken down into segments defined by a set of high-level maritime corridors or waypoints. For example, a voyage from Rotterdam to Singapore may involve traversing the English Channel, North Atlantic Ocean, Mediterranean Sea, Suez Canal, Red Sea, Indian Ocean, and the Malacca Strait. These segments form the skeleton of the overall route, capturing the general direction and major waypoints.

Once this high-level route is established, pathfinding at finer spatial resolutions can be applied between each pair of successive waypoints. These local searches operate over relatively small regions, thus they remain computationally efficient despite the higher resolution at which they operate. Meanwhile, the broader segments of the voyage across the open ocean can be planned using coarser resolutions. In doing so, the hierarchical approach preserves solution quality while significantly reducing computational overhead.

This paradigm also enables the pre-computation and storage of routes and distances between commonly used waypoint pairs. These precomputed segments can be quickly retrieved during runtime, which improves the performance further especially in large-scale or repeated route planning scenarios.

Pre-computation of intra-grid cell path costs

Pathfinding on grids typically involves routing between the centroids of navigable cells. While this approach is effective at fine spatial resolutions, it becomes increasingly inaccurate at coarser scales. In hierarchical pathfinding, where the algorithm operates on lower-resolution grids to reduce computational overhead, each cell spans a substantial area and is often too coarse to support

TABLE 2: Route lengths and runtimes for two routes at various resolutions. A^*_{QT} refers to A* operating on quadtree grids.

Route	Method	Resolution	Route length (nm)	Runtime (ms)
Piraeus, Greece	A^*_{QT}	L12	11,475	330
– Jeddah, Saudi Arabia	A^*_{QT}	L16	1,336	8,611
Shanghai, China	A^*_{QT}	L12	13,859	241
– Rotterdam, Netherlands	A^*_{QT}	L16	10,754	137

accurate estimation of travel costs. The effective distance required to traverse a cell can vary considerably depending on where a vessel enters and exits along its boundary, particularly in regions with complex coastlines or irregular terrain (see Figure 3).

To address this, we define designated entry and exit points, referred to as *portals*, along the edges of each grid cell. These portals are placed at regular intervals along the cell boundaries. Depending on whether a portal lies within navigable waters or on land, it is retained or pruned accordingly. Portals serve as the entry and exit points between adjacent cells. For each pair of portals within a cell, we precompute and store the shortest path and associated travel cost, taking into account local navigational constraints and obstacles.

This strategy effectively transforms each coarse grid cell into a local subgraph, where the portals are vertices and the precomputed intra-cell routes are weighted edges. During hierarchical pathfinding, the global routing algorithm operates on a higher-level graph composed of portals, enabling efficient routing across large areas while preserving accuracy through the use of precomputed intra-cell routes from local subgraphs. Once a high-level path is found, detailed trajectories between portals can be reconstructed using the precomputed subpaths, ensuring both computational efficiency and navigational accuracy.

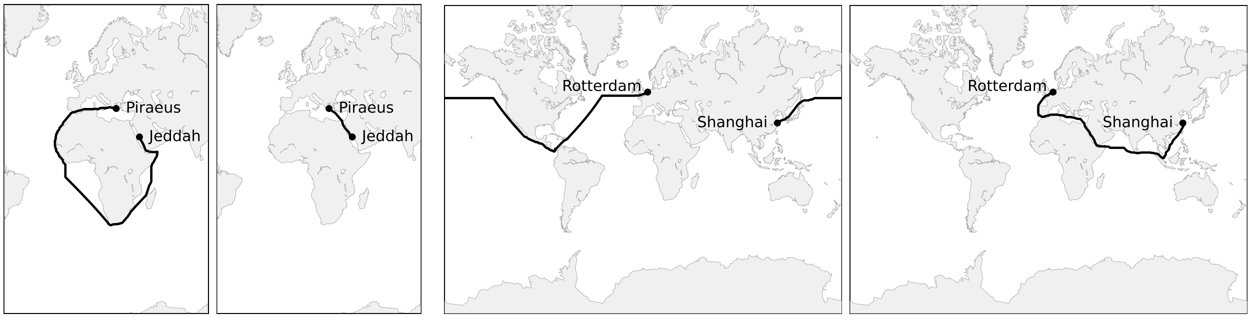
RESULTS AND DISCUSSIONS

In this section, we evaluate the performance of different pathfinding strategies in terms of both route quality and computational efficiency. Specifically, we compare regular pathfinding on uniform grids versus quadtree-based grids, followed by a comparison between regular and hierarchical pathfinding methods on quadtree grids.

The experiments are designed to assess how each approach performs across varying spatial resolutions and geographic contexts. Route quality is measured by the total travel cost, while runtime is measured as the time required to compute a route between given origin-destination pairs.

Routing performance on various spatial resolutions

As previously discussed, achieving satisfactory routing performance through narrow waterways and accurately navigating complex terrains requires the use of a high-resolution graph of the navigable waters. Figure 4 illustrates the difference in computed routes between two pairs of ports using grids with resolutions L12 and L16, respectively. With the coarser L12 grid, the algorithm fails to route through the Suez Canal, as the waterway is not adequately captured at this resolution (see Figure 1 for a visualization of the grids overlaid on the canal). As a result, the routes detour



(a) Piraeus - Jeddah (b) Shanghai - Rotterdam

FIGURE 4: Routes computed using quadtree grids of varying resolutions (L12 and L16). The algorithm fails to route the vessel through the Suez Canal in the coarser grid (L12) due to misrepresentation of the canal (see Figure 1).

FIGURE 5: A route originating from the Port of New York, obtained using grids of varying resolutions (L12, L16, L19). Finer grids capture complex terrain features more accurately, resulting in more precise estimates of travel costs.

via the Cape of Good Hope and the Pacific, leading to significantly higher travel costs.

More importantly, coarse grids can produce unrealistic or invalid routes through complex terrains. As shown in Figure 5, finer grids are essential for accurately modeling such regions and avoiding errors caused by oversimplified representations. However, using finer grids also means computing the shortest path through a much larger graph, which leads to increased runtime (see Table 2). Fortunately, the increased runtime can be mitigated through a combination of variable-sized grids and a hierarchical routing strategy.

Variable-sized grids and hierarchical routing

Through the use of variable-sized grids, we adapt the grid resolution to the terrain. That is, finer grids are used near complex terrains that require higher accuracy, while coarser grids are used for the open waters. We implemented a quadtree structure for this purpose. We have previously established that the quadtree grid significantly reduces the number of vertices in the graph (see Table 1), which in turn lowers computational runtime. We have also demonstrated this in the

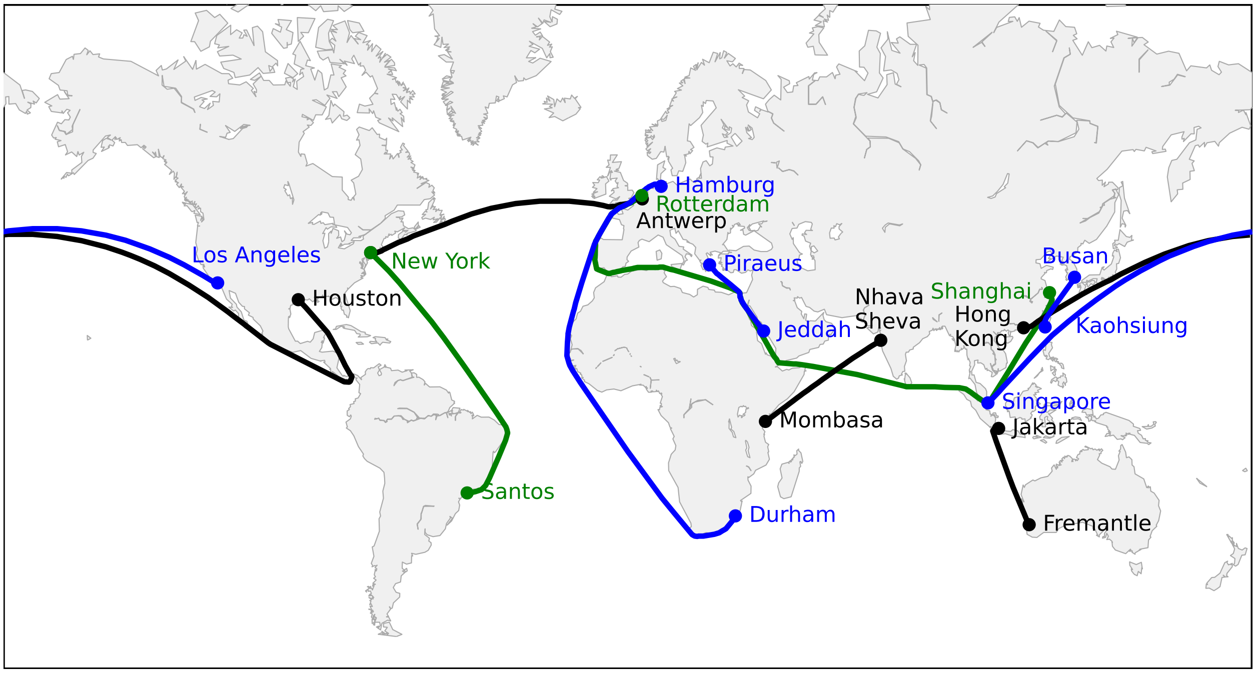


FIGURE 6: Routes generated between all ten pairs of ports used in the evaluation. The routes are computed using hierarchical A* on a quadtree grid. For clarity, the colors of the origin and destination ports correspond to those of their respective routes.

previous section through the Piraeus-–Jeddah and Shanghai–Rotterdam routes (see Table 2).

With Hierarchical A*, a high-level route is initially generated with coarse nodes, after which each segment is refined through finer grids. This approach also allows us to precompute the shortest paths within each coarse grid, which are subsequently used during high-level routing. Compared to running regular A* on the quadtree grids, and especially against A* on uniform grids, this hierarchical strategy can drastically reduce runtime. In this section, we demonstrate the effectiveness of both quadtree grids and hierarchical routing through ten select routes of different lengths across various geographical regions.

From Table 3, we observe a consistent reduction in both route length and average runtime across all evaluated routes when moving from A* on a uniform grid (A^*_{U}) to A* on a quadtree grid (A^*_{QT}), and further to hierarchical A* with a quadtree grid (HA^*_{QT}). Notably, the use of quadtree grids alone reduces runtime by over 99%, with an additional 85.4% reduction achieved through hierarchical routing.

The route quality also improves consistently from A^*_{U} to A^*_{QT} and HA^*_{QT}. This improvement stems from the use of A* on grid-based graphs. A* is restricted to movement between the centroids of grid cells, which results in jagged paths with more turns, especially when traversing many small cells. However, the minute gain in solution quality shown in this exercise is insignificant as all methods can benefit from post-processing techniques such as path smoothing or waypoint reduction.

Interestingly, runtime does not correlate directly with route length, but rather with the complexity of the terrain traversed. For example, the Nhava Sheva-–Mombasa route, though 27% longer than the Fremantle—Jakarta route, has only one-tenth the runtime. This stark dif-

TABLE 3: Route lengths and runtimes for A^*_U, A^*_{QT}, and HA^*_{QT} across selected global maritime trade routes at L19 resolution. ∗ marks routes where the algorithm ran out of memory and returned no result.

Route	Method	Route length (nm)	Runtime (ms)
Busan, South Korea	A^*_U	927	520,646
– Kaohsiung, Taiwan	A^*_{QT}	918	846
	HA^*_{QT}	904	95
Piraeus, Greece	A^*_U	1,350	782,666
– Jeddah, Saudi Arabia	A^*_{QT}	1,337	1,147
	HA^*_{QT}	1,318	56
Fremantle, Australia	A^*_U	1,865	3,885,797
– Jakarta, Indonesia	A^*_{QT}	1,854	629
	HA^*_{QT}	1,762	163
Nhava Sheva, India	A^*_U	2,522	4,702,724
– Mombasa, Kenya	A^*_{QT}	2,517	67
	HA^*_{QT}	2,405	40
Antwerp, Belgium	A^*_U	∗	∗
– New York, USA	A^*_{QT}	3,410	8,276
	HA^*_{QT}	3,277	505
New York, USA	A^*_U	∗	∗
– Santos, Brazil	A^*_{QT}	5,068	34,791
	HA^*_{QT}	4,918	3,345
Hamburg, Germany	A^*_U	∗	∗
– Durban, South Africa	A^*_{QT}	7,426	46,462
	HA^*_{QT}	7,179	2,674
Singapore	A^*_U	∗	∗
– Los Angeles, USA	A^*_{QT}	7,930	22,867
	HA^*_{QT}	7,658	2,341
Shanghai, China	A^*_U	∗	∗
– Rotterdam, Netherlands	A^*_{QT}	10,757	74,779
	HA^*_{QT}	10,421	7,293
Houston, USA	A^*_U	∗	∗
– Hong Kong, China	A^*_{QT}	11,122	240,017
	HA^*_{QT}	10,731	6,695

ference arises from the latter route's passage through the complex geography of the Indonesian Archipelago. Similarly, transoceanic routes (e.g., Singapore-–Los Angeles and Antwerp–New York) are computationally cheaper than coastal routes like New York—Santos. This, again, is due to the simpler terrain over open ocean and the efficiency of variable-sized grids in such regions.

The results demonstrate that adaptive quadtree grids offer a practical and efficient alternative to uniform-resolution grids for global vessel routing. By concentrating high resolution only where needed, the quadtree approach reduces graph size and significantly speeds up routing queries without sacrificing accuracy. Although this study focuses on distance-based routing with static costs, the quadtree framework can be naturally extended to incorporate dynamic environmental factors such as weather conditions, making it a robust foundation for operational route planning systems. The performance of hierarchical routing can also be further enhanced by introducing shortcuts between nodes in regions where routes are constrained or well-established, such as narrow channels or areas governed by traffic separation schemes near ports.

CONCLUSION

We have presented an adaptive quadtree-based grid generation method for global vessel routing that selectively refines spatial resolution based on geographic complexity. The resulting graphs are significantly smaller than uniform-resolution grids, reducing memory usage and computation time while maintaining high routing accuracy. Hierarchical routing further improves efficiency through a two-stage search. Our experiments confirm the benefits of the quadtree grids and hierarchical routing on ten routes of varying distances and complexity. These findings highlight the potential of quadtree grids and hierarchical routing for maritime route planning and lay the groundwork for future extensions to incorporate time-varying costs and environmental data.

ACKNOWLEDGMENTS

This work was supported by the "Maritime Artificial Intelligence (AI) Research for Shipping Disruption Evaluation, Launch Boat Optimisation and Digital Testing of Vessel Predictive Maintenance" funded by the Singapore Maritime Institute (SMI) under Grant SMI-2025-MTP-02. Any opinions, findings, conclusions and/or recommendations expressed in this report are those of the researchers and do not reflect the views of SMI.

REFERENCES

1. *Convention on the International Regulations for Preventing Collisions at Sea, 1972 (COLREGs)*. https://www.imo.org/en/About/Conventions/Pages/COLREG.aspx, ????, accessed: 2025-7-2.
2. *Climate impact of shipping*. https://www.transportenvironment.org/topics/ships/climate-impact-shipping, 2024, accessed: 2025-7-25.
3. *EEXI and CII - ship carbon intensity and rating system*. https://www.imo.org/en/mediacentre/hottopics/pages/eexi-cii-faq.aspx, ????, accessed: 2025-7-25.
4. *EU Emissions Trading System (EU ETS)*. https://climate.ec.europa.eu/eu-action/eu-emissions-trading-system-eu-ets_en, ????, accessed: 2025-7-25.
5. Grifoll, M., C. Borén, and M. Castells-Sanabra, A comprehensive ship weather routing system using CMEMS products and A* algorithm. *Ocean Eng.*, Vol. 255, No. 111427, 2022, p. 111427.
6. Mannarini, G., D. N. Subramani, P. F. J. Lermusiaux, and N. Pinardi, Graph-search and differential equations for time-optimal vessel route planning in dynamic ocean waves. *IEEE Trans. Intell. Transp. Syst.*, Vol. 21, No. 8, 2020, pp. 3581–3593.
7. Kurosawa, K., Y. Uchiyama, and T. Kosako, Development of a numerical marine weather routing system for coastal and marginal seas using regional oceanic and atmospheric simulations. *Ocean Eng.*, Vol. 195, No. 106706, 2020, p. 106706.
8. Samet, H., The quadtree and related hierarchical data structures. *ACM Comput. Surv.*, Vol. 16, No. 2, 1984, pp. 187–260.
9. Jeong, D.-G., M.-I. Roh, I.-C. Yeo, K.-S. Kim, and J.-S. Lee, A route planning method for small ships in coastal areas based on quadtree. *Int. J. Nav. Archit. Ocean Eng.*, Vol. 17, No. 100647, 2025, p. 100647.
10. Choi, G.-H., W. Lee, and T.-W. Kim, Voyage optimization using dynamic programming with initial quadtree based route. *J. Comput. Des. Eng.*, Vol. 10, No. 3, 2023, pp. 1185–1203.
11. Botea, A., M. Müller, and J. Schaeffer, Near optimal hierarchical path-finding. *J game dev*, Vol. 1, 2004, pp. 1–30.