https://doi.org/10.1140/epjds/s13688-022-00331-z

EPJ Data Science
a SpringerOpen Journal

REGULAR ARTICLE Open Access

A path-based approach to analyzing the global liner shipping network

Timothy LaRock[1], Mengqiao Xu[2*] and Tina Eliassi-Rad[1,3]

*Correspondence:
stephanie1996@sina.com
[2]School of Economics and Management, Dalian University of Technology, Dalian, China
Full list of author information is available at the end of the article

Abstract

The maritime shipping network is the backbone of global trade. Data about the movement of cargo through this network comes in various forms, from ship-level Automatic Identification System (AIS) data, to aggregated bilateral trade volume statistics. Multiple network representations of the shipping system can be derived from any one data source, each of which has advantages and disadvantages. In this work, we examine data in the form of liner shipping service routes, a list of walks through the port-to-port network aggregated from individual shipping companies by a large shipping logistics database. This data is inherently sequential, in that each route represents a sequence of ports called upon by a cargo ship. Previous work has analyzed this data without taking full advantage of the sequential information. Our contribution is to develop a path-based methodology for analyzing liner shipping service route data, computing navigational trajectories through the network that both respect the directional information in the shipping routes and minimize the number of cargo transfers between routes, a desirable property in industry practice. We compare these paths with those computed using other network representations of the same data, finding that our approach results in paths that are longer in terms of both network and nautical distance. We further use these trajectories to re-analyze the role of a previously-identified structural core through the network, as well as to define and analyze a measure of betweenness centrality for nodes and edges.

Keywords: Complex networks; Network representation; Sequential patterns; Path data; Maritime economics; Liner shipping

1 Introduction

Maritime container shipping facilitates trade and logistics at the global scale. This global shipping system can be modeled as a complex network, with ports as nodes[1] and edges between them representing flows of shipping [1–11]. Analyzing this network can provide insight into factors important to maritime economists and shipping industry experts, such as connectivity, efficiency, and robustness of the maritime shipping network, and thus the network of global trade.

[1] We will use the words "port" and "node" interchangeably throughout this paper.

The construction of the network, particularly the choice of connections between ports, depends first and foremost on the type of data available. Many studies have used data from vessel tracking based on Automatic Identification System (AIS) data that provides fine-grained trajectories of individual vessels moving between ports [2, 7]. For example, Kaluza et al. [2] constructed port-to-port shipping networks from AIS data based on the type of cargo (bulk, oil, container) and analyzed properties of each of these networks, including distributions of (weighted) degree and clustering, the actual trajectories of ships through the network, and motif analysis. They found substantial differences between how ships carrying different cargo navigated the network, suggesting implications for the spread of invasive species through ship ballasts. Researchers have since built on their analysis, using complex network models of AIS data to study invasive species transfer [7, 12].

The availability and granularity of AIS data makes it a valuable resource, but it does have a few drawbacks. First, it requires substantial effort to collect and clean. Second, AIS data does not retain information on the unique property of liner shipping: vessels move on fixed liner service routes (which generally include source and target ports with multiple intermediaries between these end-points) [13]. Although AIS data does contain information about the sequence of ports visited by a vessel, this data alone cannot give precise information about which ports are on the same route, since a ship may be redeployed from one route to another [11].

In this work, we analyze liner shipping service routes designed by container shipping companies and curated by Alphaliner, one of the largest proprietary shipping databases in the world.[2] Each route is an ordered sequence of ports called on by shipping vessels (see Fig. 1). Each of these routes can be conceptualized as a *walk* through the port-to-port shipping network. Although this data is less granular than individual ship tracking with AIS data, it remains a valuable resource for understanding patterns in global container shipping [9, 11]. Henceforth, we will refer to the data used in this study as "the service route data".

Different network representations can be constructed from this data. Our work compares representations and methods used in previous analyses with new methods that are designed to account for sequential or pathway patterns inherent to service route data. Analysis of sequential patterns in complex networks is sometimes called *higher-order network analysis*. Higher-order generally refers to interactions between nodes in a network that include more than two nodes at a time [14]. These interactions may be unordered, as in hypergraphs [15] and simplicial complexes [16], or, as in our case, the interactions may be ordered, as in higher-order Markov models [17, 18]. For example, Saebi et al. [12] leveraged higher-order sequential information from AIS data to improve prediction of invasive species movements [12, 19]. The service route data we study here is inherently sequential, since we know not only which ports are connected in dyads, but also which ports are visited as intermediaries between pairs of ports that do not have direct connections. The availability of this pathway information motivates the path-based approach we take in this study.

We develop a path-based methodology for liner shipping service route data that refines and expands our understanding of global container shipping. Building on previous work

[2] Alphaliner: https://www.alphaliner.com/.

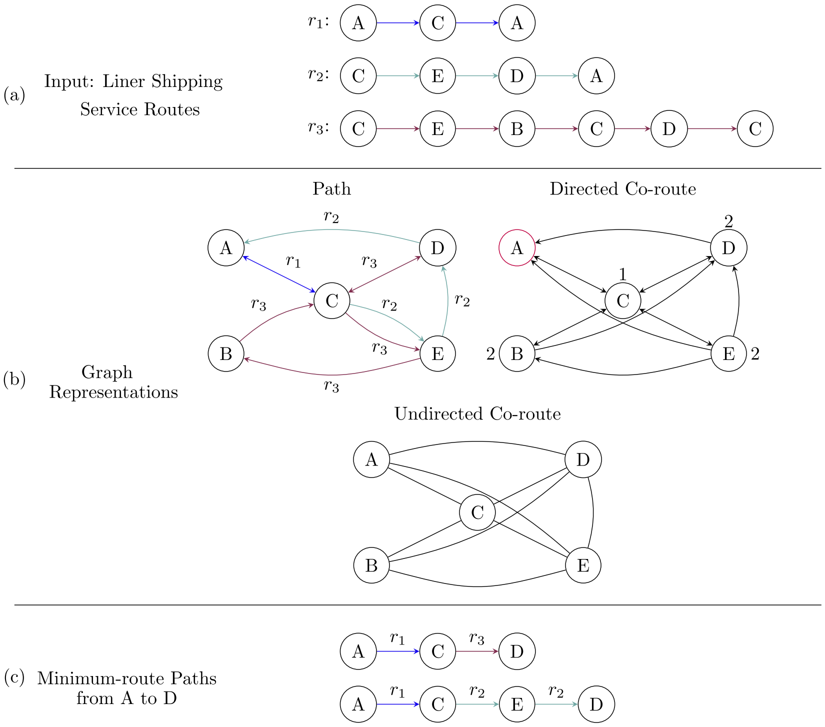


Figure 1 Examples of the three representations of liner shipping service route data studied in this paper. The input data is three routes, labeled r_1, r_2, and r_3, visiting the ports A, B, C, D, E. The path graph is the traditional directed network representation of the routes, where an edge exists from u to v if the edge appears in at least one route. We also add parallel edges for every route in which the edge exists (or equivalently keep a set of route labels for each edge). This graph represents how ships and cargo can move through the network. The directed co-route graph is also a directed graph, but an edge exists from port u to port v if in at least one route port v appears in any succeeding ports of call after port u. The length of the shortest path between any two pairs of nodes in the co-route graph is the minimum-route distance (distances from A shown in (**b**)). In the undirected co-route graph, every route is made into a clique, or fully connected undirected graph. This representation was used for service route data in previous work [11], emphasizing that cargo transportation between any two ports in a same route can be realized by one single vessel. All minimum-route paths between A and D, which require two routes and do not allow any port to appear more than once, are shown in (**c**)

that analyzed this type of data using complex networks [9, 11], our contribution is to interpolate and analyze the set of *minimum-route paths* through the routes. A minimum-route path between ports *s* and *t* is a path that starting from *s* reaches *t* using the fewest number of *transshipments* or *transfers* between shipping routes, an objective that corresponds to industry practice [20, 21]. Transshipment operations are expensive and time consuming because they require unloading, storing, and re-loading cargo containers at intermediate ports. Therefore, minimizing container transshipment improves the overall efficiency of the global liner shipping network in terms of reducing costs and delivery times [13]. Transshipments also increase risk of cargo damage, and risk of missing connections (due to the uncertainty in vessel arrival delay). Hence, having fewer transshipments to deliver a container is preferred by both liner shipping companies and their customers.

The problem of obtaining paths with the minimum cost of delivering a container in the service network of a shipping company is referred to as the *container routing problem*. Solutions to this problem are mainly based on integer programming optimization where graphs are used to represent the container flows in liner shipping networks [13, 21, 22].

The two conventional representation strategies are flows over edges of the physical network (i.e., sailing edges of consecutive ports on shipping routes) and flows over full origin-to-destination paths. Insights from our path-based approach could potentially be integrated into integer programming approaches for effectively solving the container cargo routing problem with any given operational constraints and business considerations.

Drawing on the techniques of complex network analysis, our work furthers our understanding of the functional structure of the port-to-port maritime shipping network. In previous studies of liner shipping service route data a network was constructed by making each route into a fully connected undirected graph, then analyzing shortest paths through this network representation [1, 11]. If the routes were all bi-directional, this representation would have the advantage that the shortest path length between any two nodes is equal to the minimum number of routes required to move between them. This is not always the case in the service route data used in these studies, making the path lengths through the network hard to interpret. Further, some meaningful paths through the network are impossible to compute using this representation since nodes that are only indirectly connected in a route are made to be directly connected. This makes analyses that rely on shortest paths through this network representation potentially inaccurate.

Our work addresses these inaccurate assumptions by incorporating route information into analysis of the global liner shipping network. Our contributions are:

- We compare three representations of service route data: the directed co-route graph, the undirected co-route graph, and the path graph (Sect. 3).
- We provide a procedure for computing *minimum-route paths* from liner shipping service route data, called *IMR*. We further provide algorithms for filtering minimum-route paths based on two factors, *redundancy* and *shipping distance*. We show that the properties of these paths differ substantially from the shortest paths used in previous analyses (Sects. 4.1 & 5.2).
- We reanalyze the role of the *structural core* of the global container shipping network defined in previous work [11]. We find that these analyses *underestimated* the role of core edges in routing through the network for all paths between peripheral ports and *overestimated* their role in the subset of these paths that passed through the core (Sect. 5.4). We also find the role of local edges was underestimated.
- We use minimum-route paths to define a modified betweenness centrality measure for both nodes and edges called *route betweenness*. We analyze how this measure differs from topological centrality measures across representations (Sects. 4.3 & 5.5).

Definitions We note a few basic definitions that will come up repeatedly to avoid confusion. A *walk* is a sequence of adjacent edges in a graph. Nodes and edges in a walk can be repeated. A walk is called *closed* if it starts and ends at the same node and *open* otherwise. A *route* is a pre-defined walk through the shipping network. A *path* is a walk that never repeats nodes, and a *shortest path* between two nodes s and t is a path starting from s and ending at t visiting the fewest possible edges. Walks and paths can be directed or undirected, whereas in this work routes are always directed. We will also use the word path to refer to *any* trajectory or sequence of edges in a network (e.g. the phrase path-based); we will note explicitly when we are using the term with its graph-theoretic meaning. We present a list of abbreviations and symbols we use throughout the paper in Table 1 for reference.

Table 1 List of Abbreviations & Symbols

Abbreviation	Meaning
IMR	Iterative Minimum Route
AIS	Automatic Identification System
TEU	Twenty-foot Equivalent Unit
$G_c = (V_c, E_c)$	Directed co-route graph
R	Set of routes $r \in R$ traversing ports p_i, e.g. $r = \langle p_1, p_2, \ldots, p_\ell \rangle$
ℓ_r	Length in edges of a route $r \in R$
$\mathrm{MR}[d, s, t]$	Minimum-route paths between ports s and t with distance d
$\mathrm{DIST}[s, t]$	Minimum-route distance between ports s and t
$D_{\max}$	Maximum minimum-route distance among all pairs
η_d	# of pairs at minimum-route distance d
p_d	Max # of minimum-route paths between any pair at distance d
$\mathbf{A}_{i,j}$	Adjacency matrix of the path graph (unless otherwise specified)
PG	Path graph
DCRG	Directed Co-route graph
UCRG	Undirected Co-route graph
$*$-betw	Betweenness in $*$ representation
$*$-deg	Degree in $*$ representation

The rest of this paper is organized as follows. In the next section, we provide a basic characterization of the service route data. In Sect. 3, we describe and compare three network representations for service route data. In Sect. 4, we present our proposed procedures for constructing minimum-route paths and using these paths to define new measures of centrality. In Sect. 5 we compare minimum-route paths with shortest paths through various network representations, revisit the analysis of the structural core of the global shipping network, and evaluate our newly defined measures of betweenness. Finally, we provide concluding remarks in Sect. 6.

2 Data

The service route data we study here is an aggregation of the liner shipping service routes on which shipping companies sent ships during the year 2015. The data is given in the form of 1622 liner shipping service routes, each of which is a sequence of ℓ ports visited by a ship on a single trip, e.g. $r = \langle p_1, p_2, \ldots, p_\ell \rangle$. For each route, we are given an estimate of the total capacity of that route in 2015 in Twenty-foot Equivalent Unit (TEU). Routes visit varying numbers of ports ℓ and we define the length of a route to be the number of legs $\ell - 1$ it takes, finding that the average length of a route is 6.9 edges. The same port can be visited multiple times in a single route, making each route a *walk* through the port-to-port network. A walk may be *closed* (if the route starts and ends at the same port) or *open* (if the route starts and ends at different ports). In this dataset, 1416 of the routes are closed and 206 are open.

Although we do not have data on specific container movements, we know that the routes represent the pre-determined movements of vessels carrying containerized cargo (i.e., merchandise that is shipped as container load units by sea). We also know that vessels cannot be simultaneously deployed on more than one service route at a time, and that the routes will not be modified for any vessel except in extreme circumstances. Vessels deployed on each route are always pre-fixed, and thus vessels cannot be simultaneously deployed on more than one service route at a time. Detailed cargo information (e.g., specific information concerning merchandise categories of the cargo loaded in the containers) is related to the market competences of shipping companies and is thus kept confidential and unavailable.

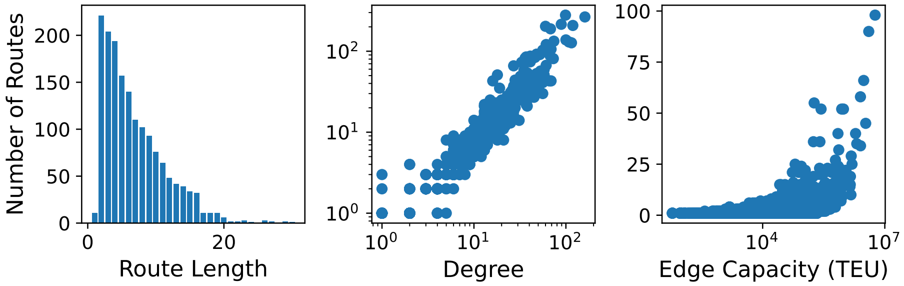


Figure 2 Statistics of the shipping routes. The left plot shows a histogram of the length of routes in terms of the number of edges. Most routes have length less than 10, but some visit as many as 30 ports. The middle plot shows the degree of a node (horizontal axis) against the number of routes the node appears in (vertical axis). The degree of a node in the path graph correlates with the number of routes it participates in and most nodes appear in a few routes, while some appear in more than 200. The right plot shows the estimated edge capacity in TEU (horizontal) against the number of routes in which each edge appears. Most edges appear in only a handful of routes, while few appear in nearly 100, and TEU capacity spans multiple orders of magnitude and correlates with the number of routes in which a port appears

Table 2 Statistics for each graphical representation of the service route data. The path graph is substantially more sparse than the other two representations

Representation	Directed?	Nodes	Edges	Mean Degree	Mean Local Clustering
Path	Yes	977	5268	5.4	0.26
Co-route (directed)	Yes	977	30,035	30.7	0.64
Co-route (undirected)	No	977	16,680	34.15	0.71

In Fig. 2 we show the distributions of route lengths (left), the degree of a node against the number of routes in which it participates (middle), and the estimated TEU capacity of an edge against the number of routes in which the edge participates (right). These distributions take similar forms, with the majority of routes being short and the number of routes each node and edge participates in being relatively small, but with tails in higher values. There are also positive correlations between the degree of a node and the TEU capacity of an edge with the number of routes in which the node or edge appears. This immediately suggests that ports and edges play different roles in the navigation of the network and that ports can be in principle separated into more central or important versus more peripheral, a common observation in shipping network analysis that is fundamental to the analyses that follow in this work [3, 11].

3 Background: three representations of liner shipping service route data

Previous work analyzing route data, including both maritime shipping and other transportation networks like public transit and railroad networks, have developed representations for the data that each have advantages and limitations. In this section we define three graph representations of route data and discuss their tradeoffs. We present examples of each representation in Fig. 1 and statistics for each representation in Table 2. All three representations include 977 ports, but the density in terms of number of edges, the average degree, and the average clustering differ substantially across the representations, as does the interpretation of the connections in the network.

3.1 Path graph

The first representation, which we call the path graph, is the standard representation of a directed and weighted network. In the path graph, an edge exists between nodes u and v if that edge appears exactly in a route, e.g. there is some route r such that u and v appear in sequence in r. In this representation we also label the edges with the routes in which they appear, equivalently formalized as many parallel labeled edges (1 per route the edge appears in), or a single set of routes as an edge attribute (with cardinality the total number of routes the edge appears in). The degree of a port u in the path graph indicates the number of unique ports (excluding parallel edges) that can be reached directly from u. The weighted degree (or *strength*) of a node u is the total number of edges the node participates in across all routes (including parallel edges). This representation is the most sparse of the three we analyze, with 5268 edges, average degree 5.4, and average local clustering coefficient 0.26.

3.2 Directed co-route graph

In the directed co-route graph, a directed edge exists between u and v if there is some *open* route r such that u appears before v in r, or some *closed* route r such that u and v both appear in r (the assumptions behind this construction are discussed further in Sect. 4.1). In this graph, the shortest path distance between any two nodes u and v represents the minimum number of routes required to reach v from u. However, shortest paths themselves through the graph may be misleading, since direct edges are drawn even where actual connections between ports are indirect. For example, in the co-route graph in Fig. 1, A-C-B is a shortest path between nodes A and C. However, the edge C-B never appears in the shipping routes. To get from A to B, a container would need to visit the edges C-E and E-B. In this representation, degree indicates the number of ports that can be reached through some other port, taking directionality into account. The strength of a node is its degree including all parallel edges representing different routes. This representation is more dense than the path graph, with 30,035 directed edges, average degree 30.7, and average local clustering coefficient 0.64.

3.3 Undirected co-route graph

In the undirected co-route graph representation each shipping route is made into a fully connected and undirected graph, i.e. a clique. If the routes being represented are bidirectional, then the shortest path length between nodes u and v in this graph reflects the minimum number of routes required to navigate between u and v. However, the shipping routes used in this and previous studies are often not bidirectional. This representation also suffers from the same problem as the co-route graph that shortest paths do not reflect actual navigation trajectories. For example, in the undirected co-route graph in Fig. 1 there is a direct edge A-D, but this edge does not appear in the shipping routes. In this representation, degree indicates the number of ports that can be reached using a single route, assuming the routes have no directionality. This is the most dense representation, with 16,680 undirected edges (33,360 directed), average degree 34.15, and average local clustering 0.71.

In the remainder of this work, we implicitly use the path graph representation as our network of interest, in contrast to some previous work on liner shipping service route data that studied shortest paths through the undirected co-route graph representation [11].

Rather than studying shortest paths, we compute paths that use the minimum number of route transfers, or *minimum-route* paths, which we define in the next section.

4 Proposed methods

In this section we describe our methodology for studying the liner shipping service routes from a path-based perspective. We define minimum-route paths and a procedure, *IMR*, for computing them, as well as additional procedures for filtering redundant and unrealistic paths. Then we use these paths to define measures of port and edge betweenness for route data.

4.1 Minimum-route paths

Intuitively, a *minimum-route path* is a path from a source port to a target port that uses the minimum number of transfers between shipping routes. We are interested in these paths because they minimize the number of times a container needs to be unloaded and reloaded at an intermediate port, which is costly in terms of time, money, and coordination. In practice there are often many minimum-route paths between a given source and target pair. These paths are at least as long as the shortest path (in edges) between the source and target ports in the path graph, and may be longer if using the shortest path would require using more than the minimum number of routes. For example, if the path A-B-C-D connects A and D using only 1 route, while the path A-E-D connects the ports using 2 routes, only the first will count as minimum-route.

Many shipping routes are closed walks, meaning they start and end at the same port. In industry practice, ships circulate on these routes regularly, meaning paths may continue from the end of the route on to the beginning. For example, in the route A-B-C-T-E-F-S-C-A, which starts and ends at the same port A, we consider the path S-C-A-B-C-T to be a valid minimum-route path between S and T that uses 1 route. Note that in these cases we allow cycles to occur in the route. There are alternative assumptions we could make that disallow cycles. We could use the path S-C-T, which only uses 1 route but also requires a transshipment, since a container traversing the path would need to be unloaded at port C, then loaded on another ship and brought to port T. This could be reasonable in some cases where the number of intermediate ports is very large, or it could be unreasonable if the number of intermediate ports is small. Alternatively, we could not allow any path from S to T using this route. This corresponds to a very strong assumption about directionality of the routes, but it misrepresents how the system operates, since routes are intentionally designed as closed walks.

We construct minimum-route paths directly from the shipping routes R as input using a procedure we call *IMR*. We build the set of paths iteratively, starting with pairs that can be connected using 1 route. For this, we loop over each route $r \in R$, checking if the route is *open*, meaning the first and last nodes are not the same, or *closed*, meaning r starts and ends at the same port. In the case where r is open, we add all of the paths between each pair of indices $i, j, i < j$. If r is closed, we add all paths between all pairs $i, j \in r$, $i \neq j$, allowing paths to continue from the end of the route to the beginning. Finally, we iterate over each minimum-route distance d, finding all minimum-route paths for pairs of nodes that require d routes. At each distance, we loop over all pairs (s, t) that are reachable using the current number of routes d. Then, we loop over all $d - 1$-route paths from s searching for any intermediate nodes w that have a 1-route path to t (by definition such a path exists). For any ports w such that a path $s \cdots w \cdots t$ exists, we record all such paths. When minimum-route paths have been computed for all pairs at distance d, we restart the while loop until all pairs have been evaluated.

Pseudocode for the *IMR* procedure described above is available in Algorithm 1 of Appendix A. We also include in Appendix A a detailed analysis of the runtime of *IMR*. Importantly, the runtime does not depend only on the structure of the network (e.g., the number of nodes and edges), but also on the number of input routes and the distribution of their lengths, the maximum minimum-route distance between any two ports in the input, as well as the distribution of the number of minimum-route paths per pair of ports.

4.2 Filtering minimum-route paths

There may be many paths between any given pair of nodes that use the minimum number of routes, and not all of these paths are equally desirable or plausible for navigation of the network. The assumption underlying the minimum-route paths is that shippers prefer to minimize trans-shipments, but other factors are also relevant to choosing between potential shipping routes. Two such factors are the existence of shorter routes that visit essentially the same set of ports and the total shipping distance of a route.

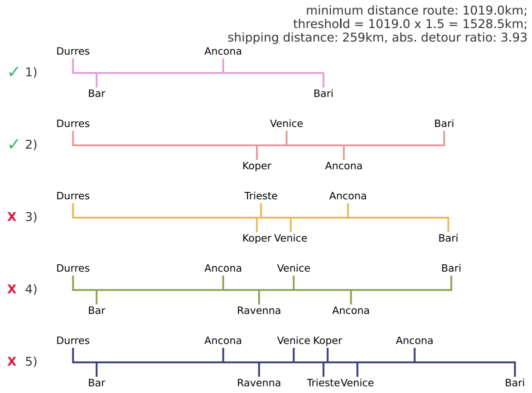
Motivated by these considerations, we filter minimum-route paths using two criteria:

1. *Redundancy:* A path is excluded if there is a shorter path in which every port in the shorter path is also visited in the longer path. Concretely, given two paths X and Y with lengths $\ell_X > \ell_Y$, we say X *subsumes* Y if $Y \subset X$. Any path that subsumes another path is called redundant and is filtered. For example, consider the longer path $X = A, B, C, D, E$ and the shorter path $Y = A, B, E$. The intersection between X and Y is all of Y, $X \cap Y \equiv Y$, meaning that $Y \subset X$ and X is redundant. In contrast, the longer path $X' = A, F, G, E$ would *not* be redundant with Y because the node B does not appear in the longer path X', meaning $Y \not\subset X'$ and thus X' does not subsume Y.
2. *Distance:* We filter paths based on distance in two ways and compare the results. The first method uses a simple threshold on the shipping distance. A path is excluded if its total shipping distance is more than a factor $\alpha \geq 1$ of the minimum distance route. Concretely, for each pair s, t we compute the total shipping distance (in km) for every path from s to t, then set a threshold using the minimum of these distances multiplied by α. Smaller α (closer to 1) will filter out more paths, since only paths with shipping distance close to the minimum will remain. If $\alpha = 1$ all paths except the minimum distance path are filtered, while $\alpha = \infty$ filters no paths. The second method uses the *detour factor* of the minimum shipping distance path with the direct shipping distance and compares with the detour factor of all other paths with the minimum-distance path (described further in Sect. 4.2.2).

We present further details of these filtering procedures below, as well as pseudocode and runtime analysis for the procedures in Algorithms 2 and 3 of Appendix A.

4.2.1 Distance threshold filtering

In the first filtering method, we compare the shipping distance of each path to this minimum shipping distance among all paths multiplied by the filtering threshold parameter α, removing any paths p such that $\textsc{dist}[p] > \alpha\textsc{dist}_{\min}$. Figure 3 shows an example of the filtering process using all of the minimum-route paths between the ports at Durress, Albania and Bari, Italy. The paths range in shipping distance from 1019 km to 1798 km,



Shipping distance	Distance > 1528.5km?	Rel. Detour Ratio	Redundant with
1019	N/A	N/A	N/A
1510	no	1.48	∅
1527	no	1.5	{2}
1539	yes	1.51	{1}
1798	yes	1.76	{1,2,3,4}

Figure 3 Example of our approach to filtering minimum-route paths. Paths 1 and 2 are kept because their shipping distances are not longer than 150% of the shortest distance, nor are they redundant with any of the other paths. Paths 3, 4, and 5 are all redundant with at least one other path, and paths 4 and 5 are additionally filtered based on their shipping distance. In this example, none of the paths would be filtered based on the detour ratio method, since the absolute detour ratio (3.93) is larger than all of the relative detour ratios

and some of the longer paths have significant overlap with shorter paths. The purpose of our filtering is to remove paths that are prohibitively long in terms of shipping distance (with respect to the minimum distance path between Durress and Bari) and those that are significantly redundant with shorter paths, since shippers are likely to simply choose the shorter of the paths. In the example, path 1 is automatically kept because it is the shortest both in terms of the number of edges used (3) and the total shipping distance (1019 km). In this example, we set $\alpha = 1.5$, meaning the shipping distance of a path must be less than 150% of the minimum, in other words we set the threshold to be $1019 \times 1.5 = 1528.5$ km. Based on this threshold path 2 is kept because its distance is less than 150% longer than the first path and unlike path 1, path 2 does not visit the port at Bar. Path 3 is acceptable based only on distance, but it is redundant with path 2 because it visits all of the same ports, but adds a stop in Trieste, Italy. The final two paths are filtered because they are both too long (151% and 176% the minimum, respectively) and redundant with at least 1 other path. In fact, path 5 is redundant with all of the first four paths.

4.2.2 Path filtering via detour factors

Using a threshold on the minimum shipping distance to filter out long paths has the advantage of simplicity, but the disadvantage that it is one size fits all, meaning the same thresholding factor α is used for every pair regardless of the distribution of shipping distances, and this parameter α must be set heuristically. This is unsatisfying because we expect these distributions to be different depending on the geographic distribution of the ports, with some quite far apart and others close together. An approach to filtering that does not require a single parameter to govern the filtering of all pairs of ports would be preferable. In this section, we develop such an approach using the *detour factor* [23] (also known as the *detour ratio*).

Given two alternate paths p_1 and p_2 between ports s and t with respective (spatial) distances d_1 and d_2, we define the detour factor between the two paths to be $\mathrm{DR}(p_1,p_2) = \frac{d_1}{d_2}$. In the canonical detour factor, d_2 represents the great-circle distance between s and t, guaranteeing that $d_1 \geq d_2$ and so $\mathrm{DR}(p_1,p_2) \geq 1$.

In this work we define two slight modifications of this usual definition. First, we define the *minimum distance detour factor* $\mathrm{DR}_{s,t}^{\min}$ to be the detour factor when p_1 is the minimum shipping distance non-redundant path between s and t and d_2 represents the shipping distance (rather than the great-circle distance) between s and t. Second, for each non-redundant path between s and t that is not the minimum shipping distance path, we define the *relative detour factor* $\mathrm{DR}_{s,t}^{r_i}$ to be the detour factor when p_1 is the path in question and p_2 is the minimum shipping distance path between s and t.

Finally, we filter a path if its relative detour factor is larger than the minimum distance detour factor, e.g. if $\mathrm{DR}_{s,t}^{r_i} \geq \mathrm{DR}_{s,t}^{\min}$.

4.3 Minimum-route betweenness

Previous work has used betweenness centrality in the port network as a proxy for measuring the importance of a port to the navigation of the system [3, 9, 11]. Betweenness centrality for a node u is defined as the sum of the proportion of shortest paths that include u between each pair of nodes (s, t) for all pairs $s \neq t \neq u$. We modify this definition by replacing the shortest paths between each pair with the set of (possibly filtered) minimum-route paths between s and t. Using this alternative set of paths defines a measure we call *route node betweenness centrality* that is based on navigation of the network using the shipping routes rather than shortest paths. Formally, route node betweenness centrality is computed as

$$rb(u) = \sum_{s,t} \frac{\sigma_{s,t}(u)}{\sigma_{s,t}},$$

where $\sigma_{s,t}(u)$ is the number of minimum-route paths from s to t that pass through the node u and $\sigma_{s,t}$ is the total number of minimum-route paths between s and t.

We can also compute *route edge betweenness centrality* following the same procedure as above except replacing nodes with edges:

$$rb(e) = \sum_{s,t} \frac{\sigma_{s,t}(e)}{\sigma_{s,t}},$$

where $\sigma_{s,t}(e)$ is the number of minimum-route paths between s and t that use the edge e.

5 Experimental results

In this section, we analyze the global liner shipping service route data using the path-based methodology set out in the previous section. We begin by evaluating the effect of the distance filtering parameter α on the minimum-route paths. Then, we compare four sets of paths constructed using different representations of the service route data: filtered minimum-route paths, shortest paths through the directed co-route graph, shortest paths through the undirected co-route graph, and shortest paths through the path graph. Next we build on previous work analyzing the *structural core* of the global liner shipping network by comparing the role of core nodes and edges using minimum-route paths. Finally, we compare our measures of port and edge importance, route betweenness centrality, with external and topological measures of importance.

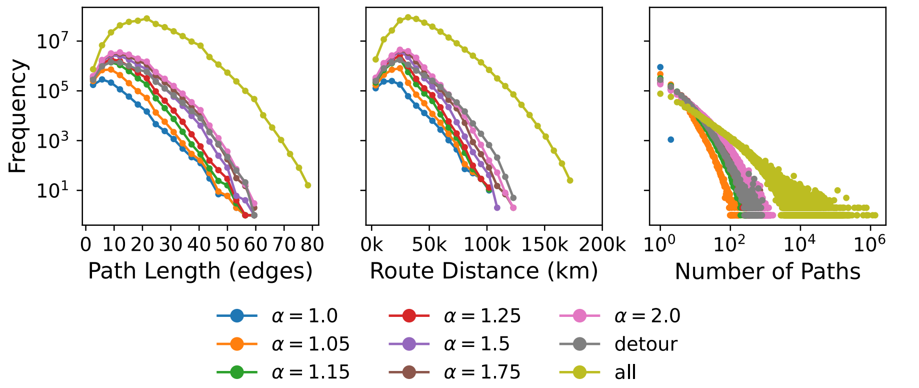


Figure 4 Outcome of filtering on minimum-route path statistics. Comparison of path length distribution (left), route shipping distance distribution (middle), and number of paths per pair (right) for all minimum-route paths against filtered paths with increasing values of the distance filtering parameter α, as well as the detour rate thresholding method. All filtering methods reduce the maximum values in each distribution and more paths are filtered when α is closer to 1. The parameterless detour rate thresholding method is more conservative than filtering using $\alpha < 1.75$. We run subsequent analyses with $\alpha = 1.15$, chosen heuristically because the results using this value fall somewhere in the middle in all distributions

5.1 Filtered minimum-route paths

We compare path length and shipping route distance statistics for all minimum-route paths and filtered paths in Fig. 4. We show results for $\alpha \in \{1.0, 1.05, 1.15, 1.25, 1.5, 1.75, 2.0\}$, as well as using the parameterless detour factor filtering. As α decreases, we see that long paths, both in terms of number of ports visited and total shipping distance, are reduced substantially. The number of paths per reachable pair is also reduced by multiple orders of magnitude after filtering, with the average number of paths per source and target pair dropping from 472 paths when including all minimum-route paths to an average of 6 paths after filtering with $\alpha = 1.15$. Using the detour factor filtering behaves similarly to filtering with the largest threshold we tested, $\alpha = 2.0$. We attribute the looseness of the filtering to the very large absolute detour factor between for some pairs of ports, which make it unlikely that the relative detour factors for the other paths will be large enough to filter.

Throughout the remainder of the analysis we use filtering threshold $\alpha = 1.15$ unless otherwise indicated. We choose this threshold because it strikes a balance between short maximum path and route lengths and filtering out almost all paths.

5.2 Comparing minimum-route paths with shortest paths

In this section we compare the filtered minimum-route paths described above with shortest paths through the undirected co-route graph (used in [11]), shortest paths through the directed co-route graph, and shortest-paths through the path graph without using route information.

Since the undirected co-route graph includes many connections that are only indirect in the other two representations, it is not possible to make an exact comparison. Out of the 900,190 pairs of ports for which we have computed minimum-route paths, only 85,781 of those pairs have at least one shortest path through the undirected co-route graph that is also viable in the path graph. There is also a mismatch in the opposite direction: since many more ports are mutually reachable in the undirected co-route graph, there are 953,552 pairs of ports with at least one shortest path between them, more than the number of

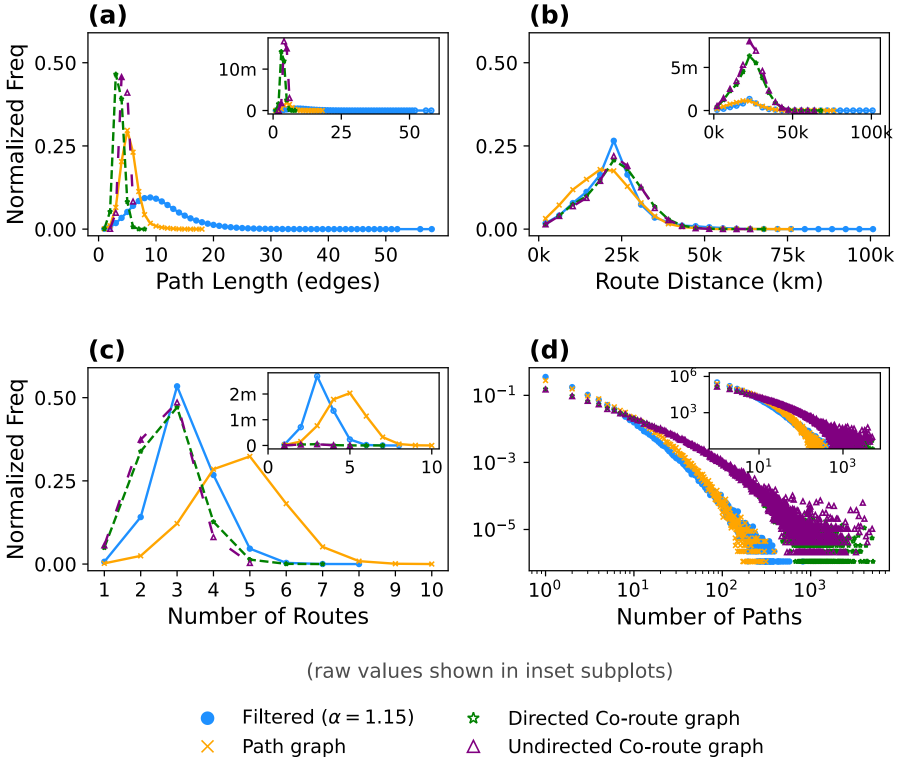


Figure 5 Filtered minimum-route paths more accurately capture how a container can move through the network based on liner shipping service routes. Path and route length statistics differ substantially between the filtered minimum-route paths (blue circles) and shortest paths through all of the path graph (orange crosses), the directed co-route graph (green stars), and the undirected co-route graph (purple triangles). Each panel shows a normalized histogram with an unnormalized version in the inset plot. Network path lengths (**a**): Minimum-route paths are much longer (mean 20 edges) than the shortest paths through the undirected (3.5) or directed (3.5) co-route graph and the path graph (mean 5.4). The mean shipping distance (**b**) for the minimum-route paths (mean 21,861.3 km) is shorter than that of the undirected (22,784.05) and directed (22,767.85) co-route graphs, but distribution for the minimum-route paths has a longer tail. The minimum-route paths are longer than shortest paths through the path graph, both in terms of mean (19,532.4) and maximum distance. The number of routes required (**c**) for the filtered paths (mean 3.2 routes) is shorter than in the path graph (4.7), where each shortest path may use a different number of routes. Only 0.3% of the shortest paths through the undirected and 0.4% of paths through the directed co-route graph are viable based on the shipping routes, but those that do correspond to viable paths use the minimum number of routes (means 2.6 and 2.7, respectively). Finally, the number of paths per pair of ports (**d**) is reduced substantially in the path graph and filtered minimum-route paths

pairs that are connected by minimum-route paths. See Table 3 for statistics on shortest paths and minimum-route paths through various co-route graphs.

This lack of alignment is impetus to take some care in explaining how we compute and report the distributions in Fig. 5. To make the potential issues concrete, there could be five shortest paths through the undirected co-route graph for a given pair. Each of these paths has the same length (by definition of the shortest path), but they may each use a different number of routes, and some may not be viable at all based on the routes. On the other hand, there may be ten minimum-route paths between the same pair, all of which use the same number of routes, but each of which is a different length. For the sake of comparison, in Fig. 5, where we compare distributions of path length, shipping route distance, and number of routes used across the three sets of paths, we (1) plot both normalized his-

Table 3 Path statistics for shortest paths through the path, directed and undirected co-route graphs, and minimum-route paths with and without filtering

Paths	# pairs	# paths	Avg paths/pair	Avg edges	Avg routes
Path graph shortest	900,190	6,289,093	7.0	5.4	4.7
Co-route (dir) shortest	900,190	30,537,099	33.9	3.5	2.7
Co-route (undir) shortest	953,552	36,565,016	38.3	3.5	2.3
Minimum-route	900,190	424,919,483	472.0	20.0	3.7
Minimum-route ($\alpha = 1.15$)	900,190	5,034,897	5.6	10.7	3.2

tograms (main plots) to compare the distributions directly and unnormalized histograms (inset) to get a sense for the differences in scale; and (2) compute one value per path between every pair, even when the values are all the same between that pair (in this sense we may describe the histograms as weighted).

In the distribution of path lengths based on the number of edges used (Fig. 5(a)), we see that the shortest paths through the directed and undirected co-route graphs and the path graphs are short compared to the filtered minimum-route paths. This is true both in terms of the average and maximum path lengths. This is important because it suggests that using shortest paths in analyzing this dataset will significantly underestimate the number of ports required for cargo to move through the shipping network.

In Fig. 5(b) we compare the distributions of route shipping distance in kilometers for each set of paths. All of the shipping distance distributions have peaks around 25,000 km, and the undirected co-route, directed co-route, and path graph have increasing maximums between 65,000 km, 70,000 km, and 76,000 km, respectively. However, the tail of the minimum-route path distance distribution is substantially longer, with maximum distance of more than 100,000 km. This again suggests that using shortest paths through any representation without accounting for the routes may underestimate the actual effort required to ship a container between some ports.

We compare the distributions of routes used per path in Fig. 5(c). As expected, the minimum-route paths use fewer routes than the shortest paths through the path graph. We note that in the case where a shortest path through the directed or undirected co-route graphs does correspond to a viable path through the routes, we know that path is minimum-route because each step away from the source port in these representations corresponds to the use of one more route [1].

Finally, in Fig. 5(d), we compare the distributions of number of paths per pair of reachable nodes. Some reachable pairs in the undirected co-route graph have upwards of 1000 shortest paths between them, while the maximum number of paths for the path graph shortest paths and minimum-route paths is an order of magnitude less.

5.3 Route importance

Since many of the same edges appear in more than one route, a minimum-route path can often be realized using multiple unique sequences of routes, each of which we call a *route sequence*. Consider the simple case of the edge C-E in Fig. 1: the edge appears in both of the routes r_2 and r_3. Now consider adding a fourth path $r_4 = E \rightarrow F$. Now to get from C to F, we can use two unique route sequences: r_2, r_4 and r_3, r_4. In this section we study the statistics of these route sequences to evaluate to what extent some routes are more important than others.

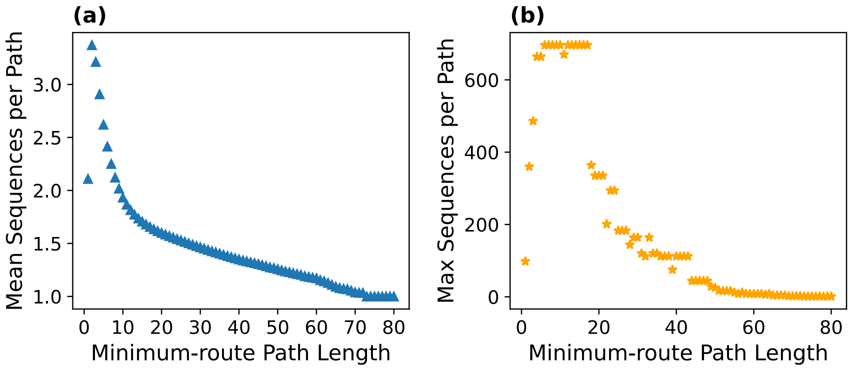


Figure 6 Many route sequences realize the same paths. Minimum-route path length (horizontal) against the (**a**) mean and (**b**) maximum number of route sequences for minimum-route paths of that length. The mean number of route sequences per path is low (less than 4) across all path lengths, but maximums reach as high as 700 sequences realizing a single path

We begin by analyzing the number of route sequences per minimum-route path. Figure 6 shows the (a) mean and (b) maximum number of route sequences per minimum-route path length. For every minimum-route path length, the average number of route sequences per path is relatively low, less than 4. However, the maximum number of route sequences is upwards of 600 for many lengths between 5 and 20, and over 100 for paths as long as 40 ports. The number of route sequences that realize a path is a function of the number of routes that connect its constitutive subpaths. For paths that can be realized using only 1 route, the number of route sequences that can realize that path is trivially the number of routes that contain it. Paths that require multiple routes are made up of multiple 1-route subpaths, and the total number of realizations is the product of the number of ways to realize each of these subpaths. For example, the minimum-route path visiting Tuticorin, Colombo, Port Kelang, Singapore, Jakarta, and finally Surabaya can be realized by 32 unique 3-route sequences. That is because there are 4 different routes that contain the edge from Tuticorin to Colombo; 8 routes that connect Colombo, Port Kelang, and Singapore; and 1 route that connects Singapore, Jakarta, and Surabaya. This implies 4 choices for the first route, 8 choices for the second, and 1 for the last, giving a total of 32 possible routes. The reason that this path has many possible realizations is that the path between Tuticorin and Singapore has many realizations, while the final leg between Singapore and Surabaya is only realizable in one way.

We show histograms counting the appearance of each route in all of the route sequences across all minimum-route paths in Fig. 7(a). For every minimum-route path, we loop over all route sequences that can realize that path. For each route, we keep a count of the total unique route sequences in which the route appears (# Sequences), as well as the total number of edges from that route used across all minimum-route paths (Total Edges). We observe that longer routes tend to appear most often, however not all of the highest count routes are long, and some are shorter than 10 ports. Given that the shipping service routes and minimum-route paths both have varying lengths, we are interested in understanding the relationship between the length of a route and how much it appears in the minimum-route paths. In Fig. 7(b), we show the length of a route against the average number of edges used when that route appears in a route sequence, computed as the total edges used divided by the number of sequences in which the route appears (the two quantities in

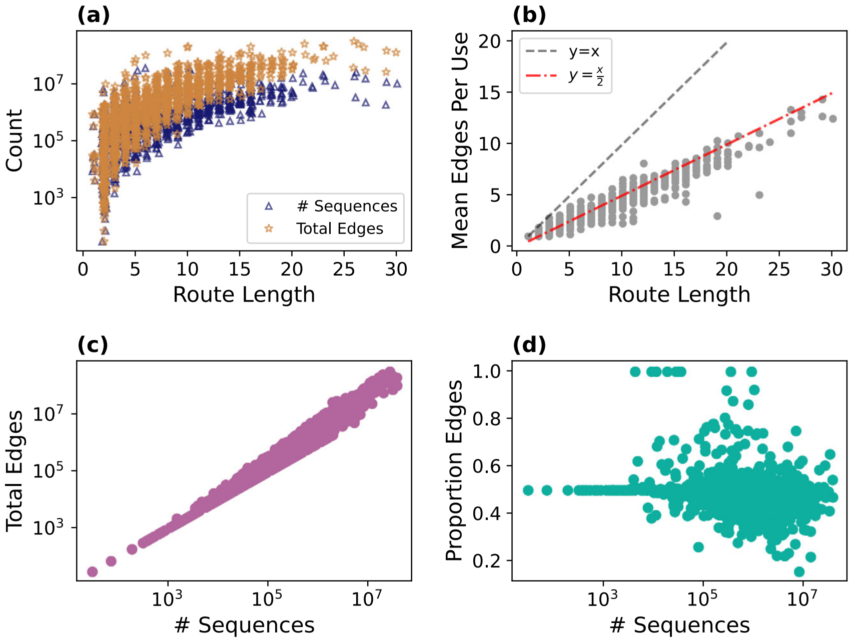


Figure 7 Longer routes use more edges on average, but all routes are not used to the same extent, and not all appearances are equal. Panel (**a**) shows the length of a shipping service route (horizontal axis) against the number of unique route sequences that route appears in (# Sequences, blue triangles), and the total number of edges from that route used across all minimum-route paths (Total Edges, orange stars). Longer service routes tend to be more highly weighted, but some shorter routes are also prevalent in realizing minimum-route paths. Panel (**b**) shows the average number of edges used per appearance in a route sequence for each service route (Total Edges divided by # Sequences). The length of a route is a natural limit on the number of edges that can be used from that route in a given path; if the points were to fall along the line $y = x$, then each time a route appeared in a route sequence we could expect most or all of its edges to be used in that minimum-route path. On average roughly half of the edges from a route are used whenever it appears in a route sequence ($R^2 = 0.92$). Panel (**c**) shows a direct relationship (in logarithmic space) between the number of sequences a route appears in and the total edges used from the route, while (**d**) shows the same quantities but with the y-axis scaled to show the proportion of the maximum possible edges. Values near the bottom indicate that few edges from a route were used with respect to the number of times the route appeared and its length, while high values indicate that each time the route is used all of its edges appear

Fig. 7(a)). We find a correlation between the length of a route and the average number of edges used, which is to be expected since there is a natural limit on the number of edges that can be used from short routes, e.g., a maximum of 1 edge can be used from a route of length 1. However, the maximum correlation in this case would be equality, meaning the whole service route was used every time it appeared in a minimum-route path realization. In our data, the average edges per use is about 50% for the longest routes (the simple model $y = \frac{x}{2}$ shown in Fig. 7(b) has coefficient of determination $R^2 = 0.92$).

In Fig. 7(c), we plot for each route the number of unique sequences in which it appears against the total number of its edges used across all paths and sequences (the quantities from Fig. 7(a)). The total number of edges scales directly in logarithmic space with the number of sequences in which a route appears. This is intuitive, since total edges increases monotonically with the number of sequences. However, because routes have varying lengths, not every appearance of a route is the same. Some long routes may contain 1 important edge that is used repeatedly, while others may be used in their entirety each time they appear. In Fig. 7(d), we rescale the vertical axis to account for the maximum possible edges from a route that could have been used given its number of sequences, dividing the total number of edges by the product of the number of unique sequences and the length of the route. We see that some routes are used in a large number of paths, but the proportion of the maximum possible edges that could have been used is less than 20%. This suggests that only sub-routes within the larger route are being used by many different paths. An example of this is a route connecting Helsinki, Finland with Szczecin, Poland. This route includes 20 port calls throughout northern Europe, including 6 ports in Finland, calls in England, Belgium, Holland, Germany, and finally Poland. This route appears in millions of minimum-route paths, but only 20% of the maximum number of edges are used. Zooming in, we find that the route is structurally important because of a few sub-routes with lengths substantially shorter than the full route length that appear in large numbers of minimum-route paths. The most frequent sub-routes of this route are the edge from Hull, UK to Antwerp, Belgium; from Helsinki to Kotka to Hamina in Finland; and the edge between Felixstowe and Hull in the UK. The least frequent sub-routes are between more peripheral ports, for example the edge between the ports at Kemi and Oulu in Finland, which appears exactly once. From this analysis we see that in some cases the importance of a route may not be determined by the importance of its start and end ports, or by the combined importance of all of its constituent ports, but by its most important sub-routes, which may make up a relatively small proportion of the entire route.

In contrast, other routes appear in many sequences and a considerable proportion of the maximum possible edges are used each time they appear. This could have two explanations. First, some routes are very short, thus are used completely each time they appear. Examples of this include the route consisting of only the edge between Busan, South Korea and Hakata, Japan, as well as the routes containing only the edge between the ports Jakarta and Belawan in Indonesia. By definition, each time one of these routes appears in a minimum-route path, the proportion of edges used is 1, since there is only 1 edge. A less trivial possibility is that some routes connect ports that are not connected in any other way, thus they are used every time those ports need to be connected. For example, a route connecting Rotterdam, Gerrmany with Hull, UK, via London and Grangemouth, is used in its entirety an order of magnitude more than its next most frequent sub-route, which is the same path but stopping short of Hull at the port in Grangemouth. This route connects Rotterdam and Hull in just 3 edges, while the two other routes in the dataset that connect these ports both require more than 10 edges and substantial intermediate detours through northern Europe.

This analysis shows that the importance of a particular route to the structure underlying the container shipping process is not just a simple function of the ports visited on the route. Instead, the role a route plays in the process is a complex and varied calculation that depends not only on itself, but also on the connections between other ports that it may facilitate.

5.4 Structural core analysis

Previous work by Xu et al. [11] classified connections between ports in the global shipping network into three categories based on whether ports involved in the connections were part of the "structural core" of ports, finding that this core plays an important role in supporting cargo transportation between peripheral ports. Using the undirected co-route graph representation, the structural core was defined by first computing a partitioning of the nodes in the network based on modularity-maximizing community detection (using the Louvain algorithm [24]). The structural core of the graph was chosen to be a set of nodes such that (1) at least one node from each of the modules was present and (2) the density of connections among the nodes in this core was relatively high. A specific set of nodes was found that satisfied these criteria (using a heuristic choice of 0.8 subgraph density): the top 37 nodes with highest value of *Gateway-ness*, a measure of the extent to which a port was connected to other ports outside of its own module, for a specific partition of the network. From here on, we will refer to all ports not in this structural core as "peripheral" ports. In this section, we compare the original analysis of the role of this structural core with an analysis using minimum-route paths rather than shortest paths through the undirected co-route graph.

With the ports making up the structural core defined, we continue following Xu et al. [11] by categorizing each edge based on whether the ports on either end are in the structural core. The original taxonomy had three categories: *core edges*, when both ports are in the structural core; *feeder edges*, when exactly one port in the core; and *local edges*, when neither port is in the core. Since minimum-route paths take directionality into account, we can split the edges in the feeder category into two categories, where an edge is an *out-feeder* if it points from a core port to a peripheral port, and an *in-feeder* if it points from a peripheral port to a core port.

Figure 8 shows reproduced results from Xu et al. [11] (left) and results computed using minimum-route paths (right). The leftmost bar in each plot shows the percent of edges in the graph (the undirected co-route and path graphs, respectively) that fall into each category. The second bar represents the percent of total shipping length, measured as the sum over all edges (u, v) of the real distance in kilometers that a vessel must travel to get from u to v, in each category. Both edge percentages (3.2% compared to 5.2%) and length percentages (6.4% compared to 9.8%) in the path graph are slightly larger than those reported in Xu et al. [11], suggesting that the role of core edges in the graph structure was underestimated in the previous work. We also report that out-feeder edges make up a larger percentage than in-feeder edges in the path graph representation.

The third and fourth bars in the left plot of Fig. 8 represent length percentages for all shortest paths between pairs of peripheral ports (third bar) and only those paths that include at least one core edge (rightmost bar). In the right plot we report the same quantities for minimum-route paths using distance threshold value $\alpha = 1.15$ (more values of α reported in Appendix B). Only 25% of shortest paths through the undirected co-route graph pass through the structural core. In contrast, 75% of the filtered minimum-route paths pass through at least 1 core edge. This difference between the paths helps explain the somewhat counterintuitive result that the role of core edges was *underestimated* for all paths between peripheral ports (16.6% vs. 24.2%), but *overestimated* for only the paths between peripheral ports that pass through at least 1 core edge (62.0% vs. 27.7%). In both cases the role of local edges was underestimated; these edges make up almost one third of the length in both sets of minimum-route paths between peripheral ports.

We also follow Xu et al. [11] by comparing the length-to-edge percentage ratios for each of the length percentages (numbers in parenthesis in Fig. 8). Overall these ratios are similar between the two representations. The exception is the role of core edges in mediating paths between peripheral ports that pass through the core, where the previously reported length-to-edge percentage ratio was 19.4, while the ratio in our analysis is 5.3.

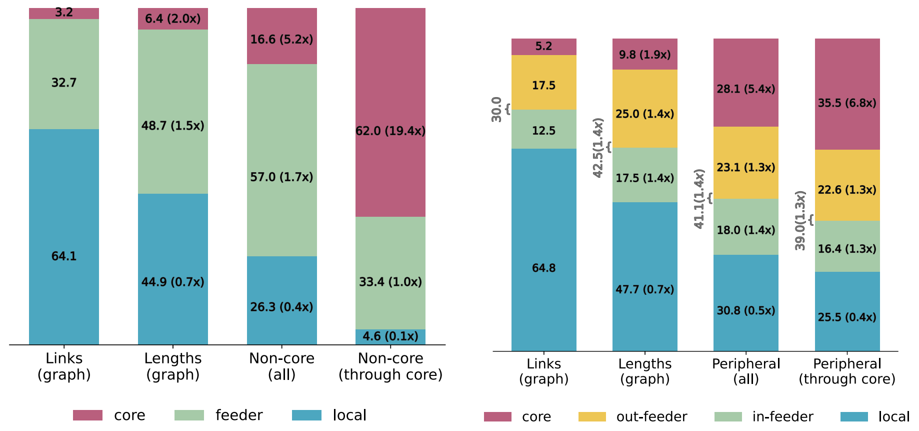


Figure 8 Percentage of edges and shipping length (real shipping distance between ports, measured in kilometers) by edge type in the original work (left) and with our updated minimum-route paths filtered with distance threshold $\alpha = 1.15$ (right). The role of the core was simultaneously *underestimated* for all shortest paths between peripheral ports and *overestimated* for only paths between peripheral ports that pass through at least 1 core edge. The first bar in each plot shows the percentage of edges in the graph that fall into each category (sum of in and out feeders in gray to the left of each bar in the right plot). The second bar shows the percentage of total shipping length for edges in each category in the graph. The numbers in parentheses are the length percentage divided by the edge percentage. The majority of edges in both representations are local (64.1% and 64.8%), but local edges make up relatively less of the total length among all edges in the graphs. The third bar in each plot shows the percentage of length by category for all paths between peripheral ports (75% of shortest paths in the previous analysis vs. 26% here), while the fourth bar shows the same quantity but only for paths that pass through at least 1 core edge (25% previously vs. 74% here). Using shortest paths through the undirected co-route graph we find that the role of the core was under-estimated for all paths (16.6% vs. 24.2%). However, the length-to-edge percentage ratios are similar for all categories (core: 5.2 vs. 4.6; feeder: 1.7 vs. 1.3; local: 0.4 vs. 0.5). Limiting to only paths that pass through the structural core, the role of the core appears to have been overestimated (62.0% vs. 27.7%) and the role of local edges was underestimated in the previous work (4.6% vs. 33.3%). In this case, the length-to-edge percentage ratio for the core is much smaller (19.4 vs. 5.3), while the feeder (1.0 vs. 1.3) and local (0.1 vs. 0.5) ratios are again similar. Out-feeder edges, which are edges that leave the structural core, make up a higher percentage of both edges and lengths than in-feeder edges, which go from outside the core to inside

The difference in estimates of the role of core edges can be explained in part by the choice of representation. When constructing the undirected co-route graph, ports that would require multiple intermediaries to reach one another based on the shipping routes are given undirected connections. Thus when shortest paths are computed, the number of intermediary nodes and edges traversed is greatly reduced, since they are bypassed by direct connections created by the undirected co-route graph construction. In contrast, in the minimum-route paths these intermediary ports must be traversed in the order they appear in routes, meaning that local edges are not avoided. The important implication of this for our analysis is that the size of the set of paths analyzed in the fourth bar changes from 25% of the paths between peripheral ports passing through the core in the original work to 75% in our analysis, which shows that core edges are indispensable in supporting cargo transportation between peripheral ports. Indeed, core edges take up an even higher percentage of the total shipping length of paths between peripheral ports that travel through the core (fourth bar, right plot) than in all paths between peripheral ports (third bar, right plot), while the difference was overestimated in previous work (left plot). By using shortest paths through the undirected co-route representation, the core was more easily avoided in the full set of paths between peripheral ports. However, this also biased the set of paths that did pass through the core towards using even more core edges. This also explains why the third and fourth bars in our analysis are more similar to one another than in the original work.

Despite the differences in the analyses, the general result from the previous analysis still holds. Our minimum-route path based analysis suggests that the structural core identified in the previous work does play an outsized role in mediating possible paths for cargo to take through the shipping network. However, we have found that quantifying the role of this core using shortest paths through the undirected co-route graph representation simultaneously biases against and toward the core edges depending on whether the paths being analyzed travel through the core at least once.

5.5 Route betweenness

In this section we evaluate *route betweenness* as described in Sect. 4.3, comparing our minimum-route path-based measures of route node and edge betweenness with topological centrality measures in the co-route and path graph representations.

We evaluate route node betweenness by measuring the correlation of the port rankings it produces with external data. We use as our baseline for comparison the top 100 ports based on container throughput downloaded from Lloyd's List Intelligence, a leading maritime shipping analyst service.[3] From this list we construct a rank vector $t = \langle 1, 2, \dots, 100 \rangle$ for the top container throughput ports, where the entry t_i corresponds to the port i with the *ith* highest throughput. Then, we compute rankings of all ports based on route node betweenness centrality, (weighted) degree and (inverse weighted) betweenness centrality in both the path and undirected co-route graphs, and the count of the number of routes in which a node appears, where we define the weight of an edge to be the total number of times the edge appears across all of the routes. For each centrality ranking, we construct a rank vector r, where the entry r_i is the ranking in the respective centrality measure for the port with the ith highest container throughput. The result is 8 rank vectors where each entry corresponds to the same port across all vectors. Finally, we compute Kendall's τ rank correlation [25, 26] between the top container throughput ranking and each centrality ranking over a sliding window increasing in rank k.

Figure 9(a) shows the results of this analysis. All 3 centrality measures correlate positively with container throughput, consistent with previous results [11]. The number of routes that a port appears in consistently has the largest rank correlation with the top 100 container throughput ports, and the strength (weighted degree) rankings are better correlated than any of the betweenness rankings. However, the correlation coefficient for the route node betweenness ranking is consistently larger than for the other betweenness measures for all values of k, and p-values on the coefficients are significant at $p = 0.0001$ after the top 30. These results suggest that when measuring port importance, our route node betweenness measure is more consistent with a measure of importance external to the network than topological betweenness centralities, while simpler measures like weighted degree or the number of routes a port participates in are better correlates than centrality.

We repeat this process again in Fig. 9(b), but this time using the total TEU capacity of all routes that a node participates in based on the data described in Sect. 2. The top 100 TEU

[3] https://www.lloydslistintelligence.com/. Note that the top 100 container ports all together account for about 80% of world's total container throughput each year. In fact, we use the top 98 ports because the ports at Ambarli and Dandong do not appear in the service route data. Further, the ports Keelung and Taipei are combined in the top 100 dataset, while they are separate in our shipping routes. Where applicable we use the minimum ranking between the two ports.

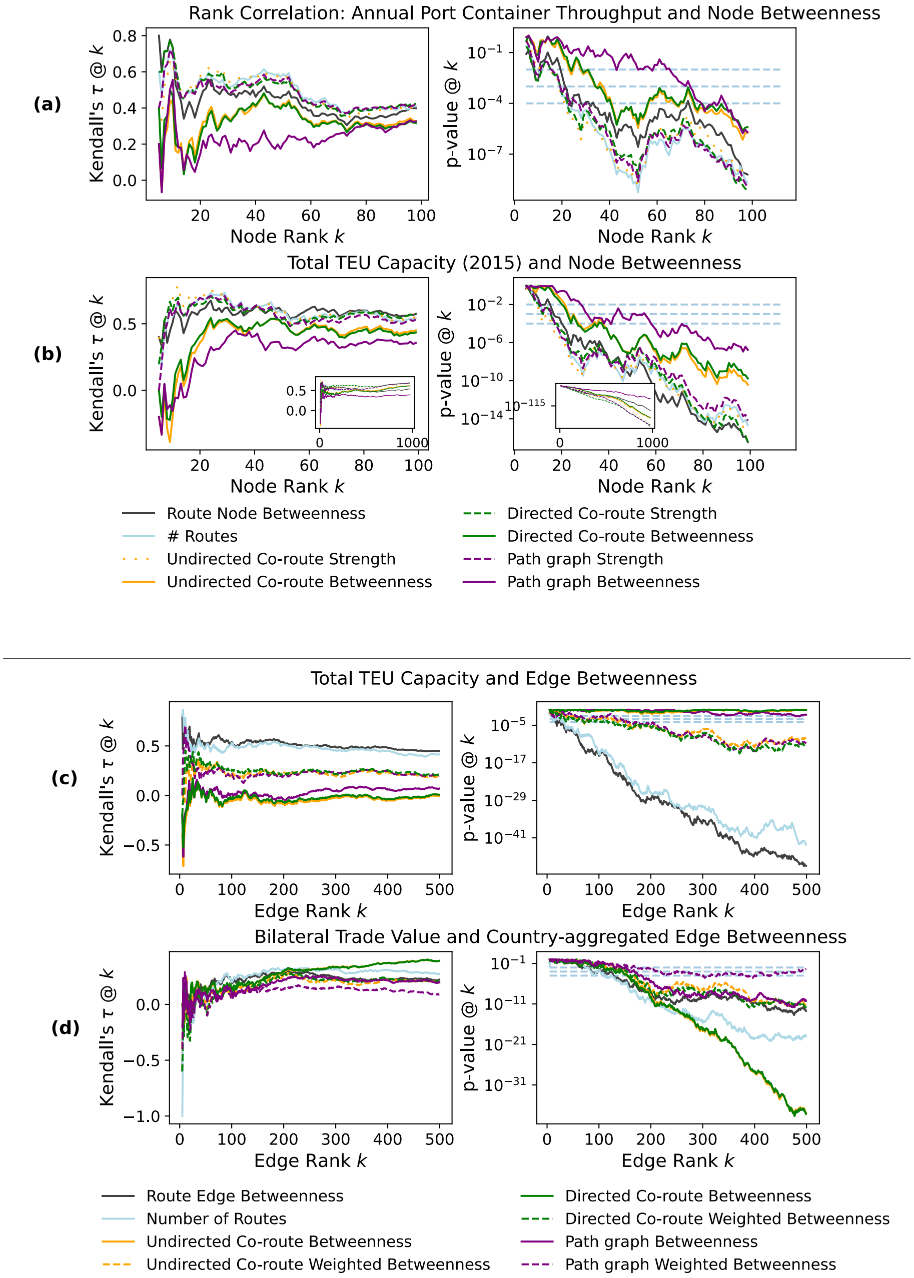


Figure 9 Rank correlations for port (a-b) and edge (c-d) importance measures. Ranking ports by number of routes they participates in has the strongest correlation with the top 100 ports by container throughput, and port rankings based on route betweenness correlate more strongly with ranking by container throughput than either topological betweenness measure. Ranking by route node betweenness correlates strongly with the top 100 ports by total TEU capacity. Correlation of edge rankings with route edge betweenness and total TEU capacity is strongly positive. Aggregating edge betweenness to the country level, betweenness in the directed and undirected co-route graphs correlates more strongly with bilateral trade than route edge betweenness or path graph betweenness. Dashed lines in the right plots indicate $p = 0.01, 0.001$, and 0.0001

Capacity port ranks are shown in the main plots, while the inset plots show correlation over all port ranks. Note that the strength rankings are determined by the total edges a node participates in over all routes, which does not include the TEU capacity information.

Results are similar to the top 100 container throughput, where the strength and number of routes measures have consistently strong correlations, and route node betweenness correlates better than the other betweenness measures.

We take a similar approach to evaluating route edge betweenness, computing the rank correlation between two external rankings and 7 edge centrality measures: route edge betweenness, (inverse weighted) edge betweenness in each of the directed and undirected coroute and path graphs. However, we must take care to properly evaluate the rankings given that edges in the directed co-route and path graph are directed, while edges in the undirected co-route graph are not. We achieve this by adding the values for the edge in both directions together, then orienting each edge across all of the rankings so the nodes are sorted alphabetically. For example, if the edges (i,j) and (j,i), $i < j$ both exist in one of the directed representations, we compute the sum of the measure of interest (e.g. betweenness) on both edges, then assign it to the single undirected edge (i,j).

The first external ranking is the sum of TEU capacity for all of the routes in which each edge appears, the same data as in Fig. 9(b). We construct a rank vector based on TEU capacity using this data. Then we construct rank vectors based on the edge centrality measures, and again compute Kendall's τ correlation between the capacity ranking and each of the centrality rankings. Results are shown in Fig. 9(c). Route edge betweenness is consistently the best correlated with edge TEU capacity. The inverse-weighted topological edge centralities correlate positively and reach low p-values by the top 500 edges, while the unweighted topological centralities hover around neutral and insignificant coefficients throughout.

The second external ranking is the bilateral trade value between countries. This analysis is of practical relevance to understanding how the structural connectivity of the global liner shipping network is associated with international trade, given the fact that liner shipping accounts for about 70% of global seaborne trade by value [11]. Since our edges are at the port level, we first aggregate the (now undirected) edge betweenness values by mapping each port to its country, then keeping a list of edge betweennness values for each pair of countries that have an edge. We then compute the rank correlation between the bilateral trade ranking and the centrality rankings, which we report in Fig. 9(d).

In this case, the directed and undirected co-route graph edge betweenness rankings are most strongly correlated with the country level trade rankings. There is an intuitive reason for the co-route graph betweenness measures to be the strongest: the country-level rankings are based on bilateral trade without specific information about who mediates relationships between countries. When the routes are transformed into fully connected and undirected graphs in the undirected co-route graph, the bilateral relationships between the countries are maintained, but the more fine-grained information about who mediates trades – in terms of maritime transportation – between the countries is lost.

Finally, we report the pairwise Kendall's τ for all importance measures in Fig. 10. As expected, route node and edge betweenness for different values of distance filtering threshold α correlate highly with one another for both nodes and edges. This, along with the results in Fig. 4, as well as Fig. 12 in Appendix B, suggest that while the choice of α does change the route betweenness values and lower α reduces shipping distances, results appear to be robust to this parameter. All pairs of node importance measures have positive and significant (at least $p < 0.01$) rank correlation coefficients. The edge betweenness rank correlations for the directed and undirected co-route graphs have neutral and even slightly

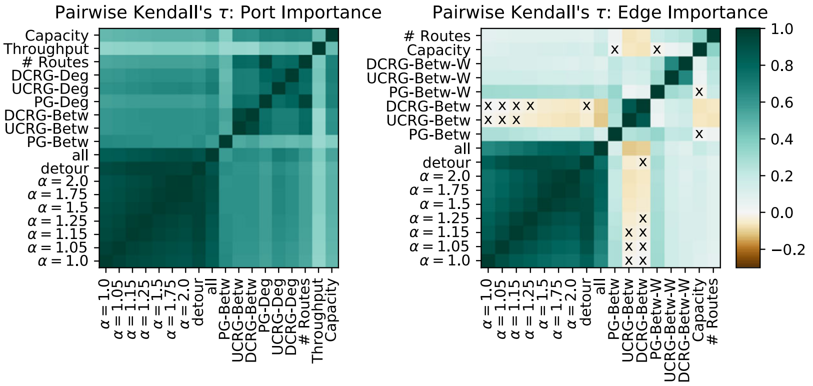


Figure 10 Pairwise Kendall's τ rank correlation for port rankings between all importance measures for nodes (left) and edges (right). "Capacity" means ranking by the sum of TEU capacity of all routes where a node or edge appears; "# Routes" means ranking by the total number of routes in which the node or edge appears at least once; and "Throughput" means ranking the top 100 ports by container throughput. "DCRG" means directed co-route graph, "UCRG" means undirected co-route graph, and "betw" corresponds to betweenness and "deg" to degree. "all" refers to the full set of minimum-route paths without filtering and "detour" refers to detour factor filtering. Route node and edge betweenness measures correlate strongly across all values of shipping distance filtering threshold α. All pairs of node importance measures have positive correlations. Edge betwenness in the directed and undirected co-route graphs are either neutral or slightly negatively correlated with other edge importance measures. Almost all p-values on the rank correlations coefficients were small; an **x** indicates that the p-value on the correlation coefficient was *not* significant at $p = 0.01$. Note that the correlations are symmetric

negative correlations with the unweighted route edge betweenness measures, suggesting that these measures are indeed capturing different kinds of edge importance.

Taken together, these results indicate that node and edge importance measures that take the service route data into account – including both our proposed route betweenness measures as well as simple counts of appearances in routes – correlate with external rankings as well as or better than measures that use shortest paths defined over the network structure. However, in some cases, such as when aggregating importance measures from ports up to countries, importance measures derived from the structure of the denser co-route graph representations may be better correlates than the route betweenness measures.

6 Conclusion

We presented analysis of liner shipping service route data using multiple network representations. We showed that the choice of representation has implications for the paths that can be inferred from the data, and that the choice of paths is important to analyzing the role of a structural core in the global maritime shipping network. Our analysis using an alternative set of paths, which we called minimum-route paths and compute using an algorithm called *IMR*, suggests that previous work underestimated the role of core edges in paths between peripheral ports and overestimated the role of core edges in the subset of paths that passed through at least one core edge. Based on this analysis we also found that previous work underestimated the importance of local edges. Despite this misestimation, the main conclusion from the previous work, that the structural core plays an outsized role in mediating navigation of cargo through the network, still follows from our analysis. Finally, we used our minimum-route paths to compute a measure of route betweenness centrality for both nodes and edges, and validated this measure against external measures of port and edge importance, finding that our measure is at least as good as other indicators for throughput and capacity based node and edge ranking, but simpler network indicators are better correlated with country-aggregated edge importance.

Our results suggest several criteria for choosing a representation when analyzing liner shipping service route data. If the research question is principally focused on dyadic trade relationships between entities, a coarser grained representation, such as the directed and undirected co-route graphs studied here, may be a reasonable and potentially advantageous representation. However, if the goal is to study the movement of cargo through the network, then either analyzing the routes themselves – as in route node or edge betweenness – or a representation that respects the directionality and direct connections in the network – as in the path graph – is likely to produce more accurate results.

In future work, results should be compared with fine-grained ship and cargo movement data that was not the focus of this study. In particular, our path-based analysis, though an improvement over the undirected co-route graph analysis, does not take the timing of ship movements into account. It is well known that the temporal ordering of edge appearances can break apparent transitivity in network dynamics [17, 18]. Future analyses should also combine liner shipping service routes with data that captures the temporal patterns of ship movements, such as AIS data, to further our understanding of temporally viable minimum-route paths by ensuring paths are *time-respecting* [27]. This could have important implications for which minimum-route paths through the network are truly viable in practice, since the temporal ordering of the trips could both limit the overall set of paths, as well as significantly alter the amount of time a path would take to realize.

Appendix A: *IMR* algorithm description, analysis, and related work

In this Appendix, we give a detailed description, including pseducode and runtime analysis, for the *IMR* procedure for computing minimum-route paths.

We iteratively construct minimum-route paths directly from the shipping routes. Algorithm 1 contains pseudocode for our proposed procedure, *IMR*. We are given as input the set of routes R. Using R, we construct the directed co-route graph representation of the routes $G_c = (V_c, E_c)$ where V_c is the set of ports and an edge (u, v) exists in E_c if either (1) there is a closed route containing both u and v, or (2) there is an open route such that u appears before v. A port t is reachable from a port s if there is at least one path that follows the directed edges in E_c from s to t. Using G_c, we compute the set of all reachable pairs using Breadth First Search from every source node in V_c and add each to the set of remaining pairs P_R. At the same time, we compute the shortest path distance $\text{dist}[s, t]$ for all pairs (s, t) in the directed co-route graph, which is the same as the minimum-route distance. This allows us to identify which pairs have minimum-route paths at each distance. We use $D_{\max}$ to denote the maximum minimum-route distance among all pairs of ports.

We build the set of paths iteratively, starting with pairs that can be connected using 1 route. For this, we loop over each route $r \in R$, checking if the route is *open*, meaning the first and last nodes are not the same, or *closed*, meaning r starts and ends at the same port. In the case where r is open, we add all of the paths between each pair of indices $i, j, i < j$. If r is closed, we add all paths between all pairs $i, j \in r$, $i \neq j$, allowing paths to continue from the end of the route to the beginning. Finally, we iterate over each minimum-route distance d, finding all minimum-route paths for pairs of nodes that require d routes. At

Algorithm 1 *IMR*(R): Algorithm for computing **I**terative **M**inimum **R**outes

```
Input: $R$ (set of shipping routes)
Output: MR (minimum route paths)
 1: Construct directed co-route graph $G_c = (V_c, E_c)$ from routes $R$
 2: Compute shortest path distances DIST$[s,t]$ in $G_c$ (BFS)
 3: $D_{\max} \leftarrow \max_{s,t}$ DIST$[s,t]$
 4: Initialize minimum-route distance $d \leftarrow 1$
    // Compute all minimum-route paths that use exactly 1 route
 5: for all routes $r \in R$ do
 6:     if $r$ is open then
 7:         Add all paths from $i$ to $j$ in $r$ s.t. $i < j$ to MR$[d,i,j]$
 8:     else
 9:         Add all paths between all pairs $(i,j) \in r$, $i \neq j$ to MR$[d,i,j]$
    // Compute paths for pairs with minimum-route distance $d > 1$
10: for $d \in 2, 3, \ldots, D_{\max}$ do
11:     for pairs $(s,t)$ s.t. DIST$[s,t] = d$ do
12:         for $w \in$ MR$[d-1, s]$ s.t. $t \in$ MR$[1, w]$ do
13:             Concatenate all paths from $s$ to $w$ with all paths from $w$ to $t$ and add to MR$[d,s,t]$
```

each distance, we loop over all pairs (s,t) that are reachable using the current number of routes d. Then, we loop over all $d-1$-route paths from s searching for any intermediate nodes w that have a 1-route path to t (by definition such a path exists). For any ports w such that a path $s \cdots w \cdots t$ exists, we record all such paths. When minimum-route paths have been computed for all pairs at distance d, we restart the while loop until all pairs have been evaluated.

A.1 *IMR* runtime analysis

The runtime of the *IMR* algorithm is the sum of the runtimes of (I) the construction of G_c from R (line 1), (II) the runtime of computing shortest path distances between all reachable pairs in G_c (line 2), (III) the runtime of the first loop (lines 5–9), and (IV) the final for loop (lines 10–13).

Steps (I) and (III) can be computed together in one loop over the full set of routes R. We let ℓ_r be the length of a given route $r \in R$. Regardless of whether a route is open or closed, the operations that compute the minimum-route paths and add edges to G_c require $O(\ell_r^2)$ time to process every pair of nodes in r. Therefore the running time is bounded by the number of routes $|R|$ multiplied by $\ell_{\max}^2$, the length of the longest route squared, resulting in the worst case running time $O(|R|\ell_{\max}^2)$.

Step (II), computing shortest path distances between all reachable pairs in G_c, can be done in $O(|V_c|(|V_c| + |E_c|))$ by running Breadth First Search (BFS) from each node in V.

Finally, step (IV) is the doubly nested for loops (lines 10–13). We defined $D_{\max}$ to be the maximum minimum-route distance among all pairs; for notational convenience we will refer to it as just D here. We note that the worst case value of D is the total number of routes $|R|$ (the case where all routes chained together in at least one ordering connect a pair of nodes that cannot be connected otherwise).[4]

We further define η_d to be the number of pairs at minimum-route distance d and p_d to be the maximum number of minimum-route paths between any pair of ports at distance d. The maximum value of η_d is $|V_c|^2 - |V_c|$ in the case where all ports are mutually reachable at the same distance d. For example, given the route A-B-C-A-B, all pairs of nodes are mutually reachable in 1 route, meaning $\eta_1 = 3^2 - 3 = 6$ corresponding to pairs A-B, A-C, B-A, B-C, C-A, C-B. In fact this maximum can only be reached when $D = 1$, since by definition edges can be navigated using exactly 1 route and so it is impossible for all ports to be connected at the same minimum-route distance $d > 1$. Thus our upper bound on η_d is loose when $d > 1$.

We also want an upper bound for the quantity p_d. An upper bound on the maximum number of minimum-route paths using d routes between a pair of nodes is the maximum number of walks between any pair. Since walks through a graph can in principle contain an infinite number of cycles, we will use the fact that the set of routes R is finite and compute a bound on the maximum number of walks between any pair of nodes using up to d routes. For a given value of d, the loose upper bound we arrive at is the maximum value of the adjacency matrix $\mathbf{A}$ of the path graph representing the routes raised to the sum of the lengths (in edges) of the d longest routes ℓ_d:

$$\max_{i,j} \mathbf{A}_{i,j}^{\ell_d}.$$

This quantity represents the maximum number of paths between any pair that use the maximum number of edges among d routes. We note that the distribution of route lengths has a tail in larger values (see Fig. 2).[5]

Each iteration of the outermost for loop (line 10) involves η_d iterations of the next for loop (line 11). In the worst case an iteration of the outer for loop requires η_{d-1} iterations of the inner for loop, each of which takes worst case time $p_d p_1$, the maximum number of paths using d routes times the number using 1 route. Thus the total running time for a particular value of d is the product of these terms: $O(\eta_d \eta_{d-1} p_d p_1)$. An upper bound on this running time is the maximum of this time over all values of d multiplied by the number of distances (iterations of the for loop in line 10)

$$\max_{d \in 1 \ldots D} \eta_d \eta_{d-1} p_d p_1,$$

which we can upper bound as

$$|V_c|^4 \Big(\max_{i,j} \mathbf{A}_{i,j}^{\ell_D} \Big)^2$$

Putting the three terms together, we have the running time

$$O\Big(|R| \ell_{\max}^2 + |V_c| \big(|V_c| + |E_c| \big) + D|V_c|^4 \Big(\max_{i,j} \mathbf{A}_{i,j}^{\ell_D} \Big)^2 \Big).$$

[4] In our dataset, D is 8 while $|R|$ is 1622.

[5] The minimum length of a route in our dataset is a single edge, the median length is 6 edges, mean length is 6.9 edges, and the maximum length is 30 edges. We further note that the number of edges used from a route in a minimum-route path is about half of the edges in the route on average (see Fig. 7(d)).

When D is 1, η_D is equal to the total number of reachable pairs, meaning the second for loop will not be entered and the last term will be irrelevant. As D grows toward its maximum $|R|$, the last term dominates the runtime. However, the upper bound approximation is worse at higher D, since the upper bound on η_d is only tight at $D = 1$, and the upper bound on p_d weakens as D grows because the approximation monotonically increases with D (e.g. $\ell_{D+1} > \ell_D$ for all D and so $A_{i,j}^{\ell_{D+1}} > A_{i,j}^{\ell_D}$) and is tightest for of a pair of nodes that is connected using all of the D largest routes in R.[6]

A.2 Filtering algorithms

We present pseudocode for the filtering procedures discussed in Sect. 4 in Algorithms 2 and 3. The input to the algorithm is mr, the data structure output by Algorithm 1; a pair of ports s and t; the minimum-route distance between the ports d; the distance filtering threshold α; and sd, a data structure containing the pairwise shipping distances between all ports. In the first outer loop we iterate over the paths p_L from longest (in terms of edges) to shortest, then in the inner loop we iterate over all paths p_s that are shorter than the current p_L. For each pair of paths, we check if $p_L \cap p_s \equiv p_s$, which indicates that the longer path subsumes the shorter path and thus should be marked redundant. If a path p_L is not redundant, we compute and store in dist$[p_L]$ its total shipping distance as the sum of the distance between all adjacent ports in the path. We also compute the minimum distance in $\textsc{dist}_{\min}$. Finally, we filter the remaining paths based on distance in one of two ways presented in the next subsection.

A.3 Filtering runtime analysis

In this section we analyze the runtime of the filtering procedure. Let p_L represent the longest path (by edges) in MR$[d,s,t]$, and let $m = |\text{MR}[d,s,t])|$, the number of minimum-

Algorithm 2 FilterPaths($\textsc{mr}$, s, t, d, α, $\textsc{sd}$)

```
Input: MR (minimum route paths), s (source port), t (target port), d (minimum-route
  distance), SD (pairwise shipping distances)
Output: F (set of paths to filter)
  $F \leftarrow \emptyset$
  // Filter longer paths if they subsume any shorter paths
  for longer path $p_L$ in $\textsc{mr}[d,s,t]$ do
      for shorter path $p_s$ in $\textsc{mr}[d,s,t]$ s.t. $|p_s| < |p_L|$ do
          if $p_L \cap p_s \equiv p_s$ then
              $F \leftarrow F \cup p_L$
              break
      // Compute shipping distance for non-redundant paths
      $\textsc{dist}[p_L] \leftarrow \sum_{i=1,\ldots,|p_L|-1} \textsc{sd}[p_L[i], p_L[i+1]]$
  Filter distances using ThresholdFilter($F, \alpha, \textsc{mr}[d,s,t]$) or DetourFilter($F, \textsc{mr}[d,s,t]$)
  Remove all paths $p \in F$ from $\textsc{mr}[d,s,t]$
```

[6] We have some evidence that this case is unlikely to appear often in real-world data. The longest minimum-route path in our dataset uses 81 edges and 4 routes, while $\ell_4 = 119$. Similarly, the longest path using $D = 8$ routes is 44 edges, while $\ell_8 = 228$.

Algorithm 3 Procedures for distance filtering

```
procedure ThresholdFilter($F, \alpha, P = \textsc{mr}[d, s, t]$)
    // Filter paths with shipping distance longer than the minimum times $\alpha$
    $\textsc{dist}_{\min} \leftarrow \min_{p \in P} \textsc{dist}[p]$
    for path $p \in \textsc{dist}$ do
        if $\textsc{dist}[p] > \alpha \cdot \textsc{dist}_{\min}$ then
            $F \leftarrow F \cup p$
procedure DetourFilter($F, P = \textsc{mr}[d, s, t]$)
    // Filter paths with relative detour factor larger than the minimum factor
    $\textsc{dist}_{\min} \leftarrow \min_{p \in P} \textsc{dist}[p]$
    $\mathrm{DR}_{s,t}^{\min} \leftarrow \frac{\textsc{dist}_{\min}}{\textsc{sd}[s,t]}$
    for path $p \in \textsc{dist}$ do
        if $\frac{\textsc{dist}[p]}{\textsc{dist}_{\min}} > \mathrm{DR}_{s,t}^{\min}$ then
            $F \leftarrow F \cup p$
```

route paths between s and t. The redundancy filtering dominates the computational complexity since it requires $O(m^2)$ time to loop over all m paths. For both distance filtering methods, we need to compute the total distance for every path, which requires $O(|p_L| \cdot m)$ time in the worst case where all paths have the longest length. We can compute the minimum $\textsc{dist}_{\min}$ at the same time. Then we need to loop over the m paths again to decide which need to be filtered. Therefore an upper bound on the running time is $O(m^2 + |p_L| \cdot m + m) = O(m^2 + |p_L| \cdot m)$. We observe that m grows much faster than p_L (see Fig. 11 in the next section), thus in practice this running time is dominated by $O(m^2)$.

The main factor in determining the running time for a specific pair of ports is the distribution of path lengths. If all paths have the same number of edges (which is unlikely), the redundancy computation can be skipped completely, since paths of the same length cannot be redundant. The more unique path lengths there are, and especially the more long paths that need to be compared with all shorter paths, the slower the computation will be. Further, the runtime of distance filtering is determined not only by the length of the longest path, but also by how many redundant paths are filtered before the distance filtering process begins, since these paths can be ignored.

A.4 Relationship between number of paths and path length

In Fig. 11 we show the relationship between the number of minimum-route paths and the maximum length among those paths for all pairs of ports. In the left plot we plot these quantities for every pair, while in the right plot we show the average number of paths for each length, with error bars shown in the inset plot. As the maximum path length increases, the number of paths per pair also increases, but at a much faster rate. This is evidence that m (number of paths) dominates p_L (maximum path length) in the runtime calculation for filtering minimum-route paths in Sect. A.3.

A.5 Related work: paths in transportation networks

Here we supplement the discussion of previous work in the Introduction section of the main text by reviewing some computational work related to computing paths through

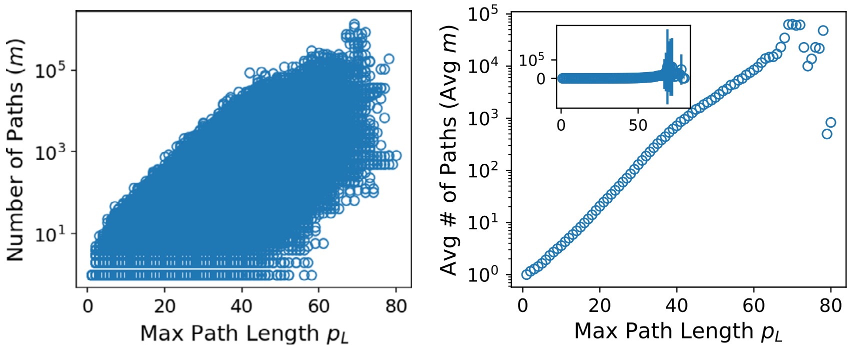


Figure 11 Maximum path length against the log of the number of paths per pair of source and target ports (left) as well as the log of the average number of paths (right, main plot) and including the amount of variation measured by one standard deviation (right, inset plot). As the maximum path length between a pair increases, so does the average number of paths between pairs with that maximum path length. An intuitive explanation for this trend is that longer minimum-route paths are concatenations of shorter minimum-route paths between intermediate source and target pairs (e.g. the loop on line 12 of Algorithm 1) that are all interchangeable. This implies that the number of paths between a pair with large maximum path length is a function of the product of the number of minimum-route paths at smaller lengths, and so the number of paths grows much more quickly than the maximum path length. However, at the highest maximum path lengths this trend does not hold. We attribute this to the fact that the very longest paths are likely to appear between ports that are not well-connected, thus there are few (perhaps only 1) viable minimum-route paths between some of these pairs, dragging the average down. Put another way: very long paths usually occur between poorly connected nodes, meaning they are more likely to be unique or few in number

transportation networks. Sequential data is the basis for many studies of transportation networks, especially in public transportation. For example, Barrett et al. presented an algorithm for solving the *label-constrained shortest paths problem* in road and rail transportation networks, taking a formal languages approach [28]; Bast et al. proposed algorithms for solving time-constrained shortest-path problems in public transportation networks [29]; Lozano et al. presented a solution to the shortest viable path problem for multi-modal networks [30]; and Lewis et al. reviewed algorithms for computing shortest-paths with vertex transfer penalties [31] (we do not have transfer times for our shipping routes).

A closely related problem is finding walks through edge-colored graphs, for example the algorithms proposed in [32]. However, typically the optimal solution is paths that use the maximum number of different colors, which in our case would correspond to using the largest number of unique routes, rather than the smallest.

The work that comes closest to our own is [33], which proposed algorithms for listing shortest paths in public transportation networks. However, the proposed algorithms assume that there are no cycles in the routes, e.g. that the routes are paths through the network, not walks. This assumption does not hold in the service route data.

Paths were constructed from public transit data for path-based analysis of the London Tube in [17, 34]. However, the method for constructing the paths was to compute shortest paths through the combined network of routes, which did not take the number of transfers into account.

Although similar to much of the above work, our study differs on a few key points. First, many transportation systems, especially public transportation, evaluated in the previous studies are based on *paths* through the network, since nodes are rarely if ever repeated in public transit routes. However, our shipping routes are not paths but *walks*, since the same ports can be visited multiple times in a single route. Second, previous work has often

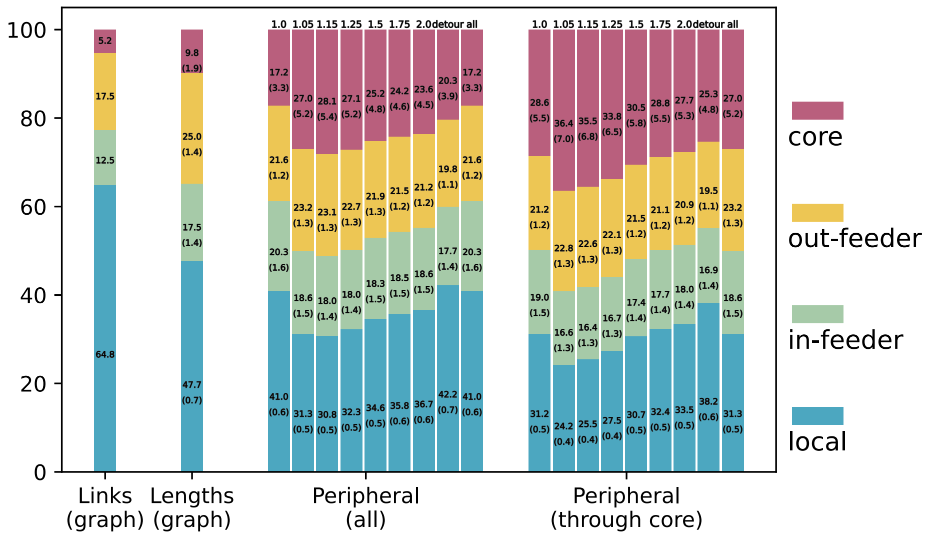


Figure 12 Percentage of edges and shipping length by edge type for various values of distance threshold α, the detour factor threshold ("detour"), as well as using the entire set of minimum-route paths ("all"). As in Fig. 8, numbers in parentheses correspond to length-to-edge percentage ratios. Results are similar for paths between all peripheral ports and paths between peripheral ports that include at least one core edge, and results are broadly similar across threshold values. Looking at all of the paths between peripheral ports, when only the minimum shipping distance path is kept ($\alpha = 1.0$), core edges account for about 17.1% of the total length, but after adding a small number of paths close to the minimum distance ($\alpha = 1.05$) the involvement of the core increases to 26.8%. As α increases, the percentage of core edges varies between 23.6% and 28.1%, while the detour threshold percentage is 20.3%. Turning to the paths between peripheral ports that pass through the core, at $\alpha = 1.0$ core edges account for about 28.5% of the total length, but at $\alpha = 1.05$ the involvement of the core increases to 35.9%. As α grows and fewer paths are filtered, the length percentage accounted for by the core decreases to 26.1% when all minimum-route paths are included, and is 25.3% using detour factor filtering

(though not exclusively) focused on shortest paths, but our work will focus on minimizing the number of route transfers.

Appendix B: Comparison of structural core results for varying thresholds

In the main text we showed the edge and length percentages for minimum-route paths using the distance filter $\alpha = 1.15$. In Fig. 12, we present results using all thresholding schemes, including detour factor thresholding ("detour") and no filtering ("all"). Regardless of the extent of filtering, we find the same result: the statistics of core edges were simultaneously underestimated and overestimated in previous work, while the statistics of local edges were underestimated; for detailed illustration, refer to Fig. 8 and its associated main text.

Acknowledgements

TL acknowledges David Liu, Adina Gitomer, and Chia-Hung Yang for conversations about computing minimum-route paths; Brennan Klein and Ryan Gallagher for advice on interpreting and visualizing results; Harrison Hartle for comments on the draft manuscript; and Leo Torres for discussion about runtime analysis. MX acknowledges lab engineer Chaoyang Bai for processing the raw data.

Funding

TL and TER are funded by in part by National Science Foundation grant IIS-1741197 and by the Combat Capabilities Development Command Army Research Laboratory through Cooperative Agreement W911NF-13-2-0045 and U.S. Army Research Lab Cyber Security CRA. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Combat Capabilities

Development Command Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes not withstanding any copyright notation hereon.
MX is funded by the National Natural Science Foundation of China (Project Number: 72101046) and the Fundamental Research Funds for the Central Universities (DUT20RC(3)046) in China.

Availability of data and materials

Raw data on world liner shipping services were provided by a third-party commercial database (Alphaliner, https://www.alphaliner.com/, one of the world's leading databases in the liner shipping industry) and were used under the license for the current study, and so are not publicly available. Data on the nautical distance between ports are publicly available in: https://www.searates.com/services/distances-time. Data on countries' international trade value and country pairs' bilateral trade value are publicly available in: https://comtrade.un.org/data. Source data are provided with this paper. We provide code for our methods and analyses, as well as some synthetic data, at https://www.github.com/tlarock/shipping.git.

Declarations

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

TL designed, implemented, and analyzed the IMR procedure; carried out all analyses; and wrote the first draft of the manuscript. MX contributed to writing and editing the manuscript; provided expertise on maritime shipping that informed all design and analysis throughout the paper; and provided all of the data, including the liner shipping service routes through an agreement with Alphaliner. TER contributed to writing and editing the paper and provided feedback on intermediate results and ideas throughout the project. All authors read and approved the final manuscript.

Author details

[1]Network Science Institute, Northeastern University, Boston, MA, USA. [2]School of Economics and Management, Dalian University of Technology, Dalian, China. [3]Khoury College of Computer Sciences, Northeastern University, Boston, MA, USA.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 25 October 2021 Accepted: 9 March 2022 Published online: 25 March 2022

References

1. Hu Y, Zhu D (2009) Empirical analysis of the worldwide maritime transportation network. Phys A, Stat Mech Appl 388(10):2061–2071. https://doi.org/10.1016/j.physa.2008.12.016
2. Kaluza P, Kölzsch A, Gastner MT, Blasius B (2010) The complex network of global cargo ship movements. J R Soc Interface 7(48):1093–1103. https://doi.org/10.1098/rsif.2009.0495
3. Ducruet C, Lee S-W, Ng AKY (2010) Centrality and vulnerability in liner shipping networks: revisiting the Northeast Asian port hierarchy. Marit Policy Manag 37(1):17–36. https://doi.org/10.1080/03088830903461175
4. Ducruet C, Zaidi F (2012) Maritime constellations: a complex network approach to shipping and ports. Marit Policy Manag 39(2):151–168. https://doi.org/10.1080/03088839.2011.650718
5. Ducruet C, Notteboom T (2012) The worldwide maritime network of container shipping: spatial structure and regional dynamics. Glob Netw 12(3):395–423. https://doi.org/10.1111/j.1471-0374.2011.00355.x
6. Ducruet C (2013) Network diversity and maritime flows. J Transp Geogr 30:77–88. https://doi.org/10.1016/j.jtrangeo.2013.03.004
7. Xu J, Wickramarathne TL, Chawla NV, Grey EK, Steinhaeuser K, Keller RP, Drake JM, Lodge DM (2014) Improving management of aquatic invasions by integrating shipping network, ecological, and environmental data: data mining for social good. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1699–1708. https://doi.org/10.1145/2623330.2623364
8. Li Z, Xu M, Shi Y (2015) Centrality in global shipping network basing on worldwide shipping areas. GeoJournal 80(1):47–60. https://doi.org/10.1007/s10708-014-9524-3
9. Xu M, Li Z, Shi Y, Zhang X, Jiang S (2015) Evolution of regional inequality in the global shipping network. J Transp Geogr 44:1–12. https://doi.org/10.1016/j.jtrangeo.2015.02.003
10. Kojaku S, Xu M, Xia H, Masuda N (2019) Multiscale core-periphery structure in a global liner shipping network. Sci Rep 9(1):404. https://doi.org/10.1038/s41598-018-35922-2
11. Xu M, Pan Q, Muscoloni A, Xia H, Cannistraci CV (2020) Modular gateway-ness connectivity and structural core organization in maritime network science. Nat Commun 11(1):2849. https://doi.org/10.1038/s41467-020-16619-5
12. Saebi M, Xu J, Curasi SR, Grey EK, Chawla NV, Lodge DM (2020) Network analysis of ballast-mediated species transfer reveals important introduction and dispersal patterns in the Arctic. Sci Rep 10(1):19558. https://doi.org/10.1038/s41598-020-76602-4
13. Wang S, Meng Q, Sun Z (2013) Container routing in liner shipping. Transp Res, Part E, Logist Transp Rev 49(1):1–7. https://doi.org/10.1016/j.tre.2012.06.009
14. Torres L, Blevins AS, Bassett D, Eliassi-Rad T (2021) The Why, How, and When of Representations for Complex Systems. SIAM Rev 63(3):435–485. https://doi.org/10.1137/20M1355896
15. Chodrow PS (2020) Configuration models of random hypergraphs. J Complex Netw 8(3):018. https://doi.org/10.1093/comnet/cnaa018

16. Battiston F, Cencetti G, Iacopini I, Latora V, Lucas M, Patania A, Young J-G, Petri G (2020) Networks beyond pairwise interactions: structure and dynamics. Phys Rep 874:1–92. 2006.01764. https://doi.org/10.1016/j.physrep.2020.05.004
17. Scholtes I (2017) When is a network a network?: multi-order graphical model selection in pathways and temporal networks. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp 1037–1046. https://doi.org/10.1145/3097983.3098145
18. Lambiotte R, Rosvall M, Scholtes I (2019) From networks to optimal higher-order models of complex systems. Nat Phys 15(4):313–320. https://doi.org/10.1038/s41567-019-0459-y
19. Xu J, Wickramarathne TL, Chawla NV (2016) Representing higher-order dependencies in networks. Sci Adv 2(5):1600028. https://doi.org/10.1126/sciadv.1600028
20. Brouer BD, Alvarez JF, Plum CEM, Pisinger D, Sigurd MM (2014) A base integer programming model and benchmark suite for liner-shipping network design. Transp Sci 48(2):281–312
21. Balakrishnan A, Karsten CV (2017) Container shipping service selection and cargo routing with transshipment limits. Eur J Oper Res 263(2):652–663
22. Jin JG, Meng Q, Wang H (2021) Feeder vessel routing and transshipment coordination at a congested hub port. Transp Res, Part B, Methodol 151:1–21. https://doi.org/10.1016/j.trb.2021.07.002
23. Yang H, Ke J, Ye J (2018) A universal distribution law of network detour ratios. Transp Res, Part C, Emerg Technol 96:22–37. https://doi.org/10.1016/j.trc.2018.09.012
24. Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008(10):10008. https://doi.org/10.1088/1742-5468/2008/10/P10008
25. Kendall MG (1970) Rank correlation methods, 4th edn. Griffin, London
26. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat İ, Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P (eds) (2020) SciPy 1.0 contributors: SciPy 1.0: fundamental algorithms for scientific computing in python. Nat Methods 17:261–272. https://doi.org/10.1038/s41592-019-0686-2
27. Scholtes I, Wider N, Garas A (2016) Higher-order aggregate networks in the analysis of temporal networks: path structures and centralities. Eur Phys J B 89(3):61. https://doi.org/10.1140/epjb/e2016-60663-0
28. Barrett C, Bisset K, Holzer M, Konjevod G, Marathe M, Wagner D (2008) Engineering label-constrained shortest-path algorithms. In: Fleischer R, Xu J (eds) Algorithmic aspects in information and management, vol 5034, pp 27–37. https://doi.org/10.1007/978-3-540-68880-8_5
29. Bast H, Carlsson E, Eigenwillig A, Geisberger R, Harrelson C, Raychev V, Viger F (2010) Fast Routing in Very Large Public Transportation Networks Using Transfer Patterns. In: Hutchison D, Kanade T, Kittler J, Kleinberg JM, Mattern F, Mitchell JC, Naor M, Nierstrasz O, Pandu Rangan C, Steffen B, Sudan M, Terzopoulos D, Tygar D, Vardi MY, Weikum G, de Berg M, Meyer U (eds) Algorithms – ESA, vol 2010, pp 290–301. https://doi.org/10.1007/978-3-642-15775-2_25
30. Lozano A, Storchi G (2001) Shortest viable path algorithm in multimodal networks. Transp Res, Part A, Policy Pract 35(3):225–241. https://doi.org/10.1016/S0965-8564(99)00056
31. Lewis R (2020) Algorithms for Finding Shortest Paths in Networks with Vertex Transfer Penalties. Algorithms 13(11):269. https://doi.org/10.3390/a13110269
32. Ferone D, Festa P, Pastore T (2019) The k-color shortest path problem. In: Paolucci M, Sciomachen A, Uberti P (eds) Advances in optimization and decision science for society, services and enterprises, vol 3. Springer, Cham, pp 367–376. https://doi.org/10.1007/978-3-030-34960-8_32
33. Böhmová K, Häfliger L, Mihalák M, Pröger T, Sacomoto G, Sagot M-F (2018) Computing and Listing st-Paths in Public Transportation Networks. Theory Comput Syst 62(3):600–621. https://doi.org/10.1007/s00224-016-9747-4
34. LaRock T, Nanumyan V, Scholtes I, Casiraghi G, Eliassi-Rad T, Schweitzer F (2020) Hypa: efficient detection of path anomalies in time series data on networks. In: Proceedings of the 2020 SIAM international conference on data mining, pp 460–468. https://doi.org/10.1137/1.9781611976236.52