Comparison of Container Routing in Liner Shipping Networks

Massimo Di Gangi, Orlando M. Belcore(✉), and Antonio Polimeni

Department of Engineering, University of Messina, Messina, Italy
{mdigangi,obelcore,antonio.polimeni1}@unime.it

Abstract. The increasing volatility of liner shipping networks, characterized by frequent capacity reallocations and schedule disruptions, calls for robust and comparable container routing models. This paper presents a systematic comparison of two approaches: the Connection Scan Algorithm (CSA) and the RAPTOR algorithm. To enable a model-independent assessment, routes are projected onto a canonical representation based on sequences of ports and liner services. Using a set of scenarios to increase network complexity, routing outcomes, computational performance, and route similarity between the two approaches are evaluated. The results show that CSA and RAPTOR exhibit greater sensitivity to network dynamics, generating structurally different routes with more transshipments, particularly in large and volatile scenarios. The proposed framework provides a unified basis for comparing heterogeneous routing models and highlights trade-offs between computational efficiency, temporal realism, and robustness, offering guidance for both strategic planning and operational analysis in modern liner shipping networks.

Keywords: container shipping · liner shipping · connection scan algorithm · RAPTOR · maritime transport

1 Introduction

Liner shipping networks constitute the backbone of global maritime transport, enabling the movement of containers across intercontinental routes [1, 2]. The planning and operational management of these networks require a detailed understanding of port management [3, 4], container routing [5, 6], transshipment processes [7, 8], and schedule synchronization [9, 10]. Container routing decisions directly affect transit times, reliability, and the utilization of maritime and terminal resources. Therefore, container routing represents a critical decision problem in maritime logistics, where containers are assigned to sequences of liner services and ports subject to service schedules, transshipment constraints, and operational requirements.

Recent developments in liner shipping operations further underscore the need for robust and temporally consistent routing models. According to recent analyses by Sea Intelligence [11], 2025 represents a turning point in ocean liner policy, distinguished by substantial changes in vessel deployments across major East–West trade lanes. This volatility is captured by the notion of operational capacity churn, defined as the sum of

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026 Workshops, LNCS 16763, pp. 527–542, 2027.
https://doi.org/10.1007/978-3-032-30530-5_33

vessel capacity entering and exiting a trade route over time, which serves as a proxy for structural instability in service deployment. Such dynamics are different from the prepandemic paradigm, in which container flows followed relatively predictable patterns. Instead, liner shipping networks increasingly resemble dynamic systems subject to short-term demand fluctuations, operational disruptions, and rapid reconfiguration of services. Routing decisions based on static or time-aggregated network representations risk overlooking critical feasibility and synchronization constraints. These developments raise fundamental questions regarding the adequacy of different network modeling approaches for simulating/designing container routing. An assessment of the adequacy of line-based representations under operational volatility is required to understand whether time-expanded formulations are needed (and more generally, time-sensitive routing approaches) to identify feasible itineraries, transfer patterns, and rerouting opportunities.

In this context, this study provides a comparison of alternative models for container routing. By designing routing solutions onto a common canonical representation, the study quantifies differences in routing outcomes, route structure, thereby offering insights into the trade-offs between modeling simplicity and operational realism in liner shipping networks.

Routing in liner shipping can be represented at different levels of detail, ranging from service-oriented models to fully time-dependent formulations that explicitly incorporate port calls and vessel schedules. Therefore, selecting an appropriate modeling approach is crucial for both strategic planning and operational decision-making, as it affects the realism of routing outcomes and the computational tractability of the problem. Network representations that explicitly account for time (such as time-expanded networks) or connection-based approaches are particularly well suited to this purpose, as they enable the identification of alternatives and the assessment of their impact on transit times and transshipments. In addition to connection-based approaches, round-based timetable routing algorithms (e.g., RAPTOR [12]) could represent a natural extension of the methodological framework considered in this study. Although a large body of literature addresses container routing and transshipment optimization [13, 14], systematic comparisons of routing approaches remain scarce, particularly with respect to:

- differences in routing decisions and transshipment patterns,
- structural similarities and divergences of the resulting routes,
- computational scalability across increasing network complexity,
- robustness under operational disruptions and re-routing scenarios.

To try to fill this gap, this research tries to answer the following questions:

- How do different temporal routing algorithms (CSA and RAPTOR) perform under increasing network complexity in liner shipping?
- What are the trade-offs between these routing approaches in terms of route quality, computational feasibility, and robustness?
- Does a unified, model-independent framework facilitate a systematic comparison and highlight key differences relevant for strategic and operational decision-making?

Routes generated using connection-based approaches and RAPTOR algorithms are projected onto a common canonical representation, enabling direct and quantitative comparison. Similarity metrics, performance indicators, and robustness analysis are

employed across a range of network scenarios, providing insight into the trade-offs between modeling goodness and computational efficiency for both strategic and operational applications. The findings are expected to be of interest for both researchers in advancing routing methods and practitioners aiming to optimize network design, enhance robustness to disruptions, and improve efficiency in container transport operations. Thus, the paper serves both as a methodological framework and as a practical guide for stakeholders.

The paper is structured as follows. Section 2 reports a brief literature review on modeling and approaches for route simulation and design. Section 3 discusses the proposed framework, while Sect. 4 summarizes routing strategies and a comparison methodology. Section 5 provides the results of an application. Finally, a section with discussion and conclusions.

2 Literature Review

Three main modeling and routing approaches are commonly encountered in literature and in practice. In line-based (or service-oriented) networks, container movements are represented as transitions between predefined service loops connecting ports. The temporal dimension is treated in an aggregated manner, with waiting times approximated through average values or frequency-based assumptions [15, 16]. In time-expanded (or diachronic) networks, the temporal dimension is explicitly embedded in the network structure by replicating port nodes for each service call. Arcs represent sailing, waiting, handling, and transshipment operations, ensuring full temporal consistency of routing decisions [17]. Event-based routing algorithms, such as the Connection Scan Algorithm (CSA), operate directly on temporally ordered service connections without requiring the explicit construction of a full time-expanded graph [12, 18]. These algorithms provide exact, time-feasible routing solutions while significantly reducing memory requirements and preprocessing effort.

Each of these approaches involves different trade-offs between modeling accuracy, computational efficiency, and scalability. Line-based models [19, 20] offer compact representations and fast computation, but rely on simplifying assumptions regarding waiting times and schedule synchronization. Time-expanded networks [21, 22] provide a detailed and temporally consistent representation of container movements, but this comes at the expense of a larger network structure. Event-based algorithms, such as CSA, constitute an intermediate paradigm, preserving temporal consistency while avoiding explicit network expansion.

Frequency-based service modeling is widely used in urban public transport to capture the stochastic and adaptive nature of passenger route choices [23–25]. In such systems, passengers can adapt their choices at stops based on service availability, leading to the formulation of hyperpaths that represent sets of feasible routes and associated choice probabilities [26–28]. Hyperpath-based models support stochastic assignment and real-time adaptive behavior. However, in the context of container shipping, the adaptive component at ports is typically not considered. Containers do not make real-time choices analogous to passengers waiting at a stop. Instead, routing decisions are defined deterministically at the strategic or tactical level, producing a single route from the point

of origin to the destination. Consequently, hyperpath formulations are not employed, and container routing is naturally modeled as a shortest-route problem on deterministic networks. This fundamental difference justifies the use of route-based rather than strategy-based representations in liner shipping applications.

Time-expanded network models explicitly incorporate the temporal dimension by replicating nodes over time. Originally developed in the context of public transport scheduling and airline operations, these models have been widely adopted for time-dependent routing and scheduling problems [29–32]. More recently, event-based algorithms, such as CSA, have emerged as efficient alternatives for timetable-based routing, offering exact solutions while avoiding the explicit construction of large time-expanded graphs. An additional motivation for considering temporally consistent routing models lies in the operational reality of liner shipping. Container routing is not a static problem: during service execution, unplanned events such as missed port calls, port congestion, or vessel delays may require re-routing of containers to alternative services or ports [3, 11]. Re-routing involves recomputing a feasible itinerary from the current state of the container onward, potentially modifying the sequence of ports and services while preserving the deterministic nature of the routing decision.

3 Modeling Framework

Given a set of ports P and liner services S, the container routing problem consists of finding minimum-cost routes between all feasible origin-destination port pairs, where costs are represented by travel time and handling time.

3.1 CSA Temporal Network Model

CSA modeling approach adopts a connection-based temporal representation of the liner shipping network and relies on the connection scan algorithm to compute feasible routes. CSA does not explicitly build a graph with temporal nodes and arcs. Instead, it operates directly on an ordered set of connections, exploiting their temporal structure to compute earliest-arrival routes. Let C denote the set of all connections consistent with S in the network. Each connection $c \in C$ is defined as a tuple:

$$c = \left(p_c^{dep}, p_c^{arr}, s_c, t_c^{dep}, t_c^{arr}\right) \tag{1}$$

where:

$p_c^{dep} \in P$ and $p_c^{arr} \in P$ are the departure and arrival port of the connection;
$s_c \in S$ identifies the liner service operating the connection;
t_c^{dep}, t_c^{arr} are the scheduled departure and arrival times.

Given an origin port $o \in P$ and a departure time t_0, CSA computes earliest-arrival times by scanning the ordered list of connections exactly once. For each port p, the algorithm maintains the earliest known arrival time $\tau(p)$, initialized as:

$$\tau(o) = t_0$$

$$\tau(p) = +\infty \forall p \neq o \tag{2}$$

During the scan, a connection c is relaxed if:

$$\tau(p_c^{dep}) \leq t_c^{dep} \tag{3}$$

in which case the arrival time at the destination port is updated as:

$$\tau(p_c^{arr}) \leftarrow \min\left\{\tau(p_c^{arr}), t_c^{arr}\right\} \tag{4}$$

By storing predecessor information, CSA also enables the reconstruction of a time-feasible route between origin and destination ports. In the CSA framework, transshipment is not explicitly modeled, it arises when two consecutive connections share the same arrival and departure port, and the departure time of the second connection is later than the arrival time of the first. Waiting times at ports are implicitly captured by the time difference between the arrival of one connection and the departure of the next feasible one. As a result, both waiting and transshipment feasibility are ensured without introducing additional structural elements. The CSA-based temporal model provides an efficient and schedule-consistent approach to container routing in liner shipping networks. It is particularly well suited for large-scale networks and extensive origin–destination analyses, where the explicit construction of a time-expanded graph may become computationally prohibitive.

3.2 RAPTOR-Based Temporal Routing Model

The modeling approach discussed in this section is based on the Round-Based Public Transit Routing (RAPTOR) algorithm and adopts a schedule-based representation of liner shipping services. Originally developed for public transport routing [12], RAPTOR computes optimal routes by iteratively relaxing services over successive rounds, each corresponding to an additional transfer. This characteristic makes RAPTOR particularly well suited for container routing, where the number of transshipments is a key operational performance indicator. Unlike graph-based formulations, RAPTOR does not explicitly construct a network of nodes and arcs. Instead, it operates directly on the ordered stop sequences of services, thereby avoiding the construction of large time-expanded graphs while preserving full temporal consistency.

Each service $s \in S$ be defined by an ordered sequence of port calls:

$$s = \left(\left(p_1, t_1^{arr}, t_1^{dep}\right), \left(p_2, t_2^{arr}, t_2^{dep}\right), ..., \left(p_{n_s}, t_{n_s}^{arr}, t_{n_s}^{dep}\right)\right), \tag{5}$$

where $p_i \in P$ is the port of the ith call and $t^{\text{arr}}{}_i$, $t^{\text{dep}}{}_i$are the corresponding arrival and departure times.

Each service is assumed to operate according to a fixed schedule within the planned horizon considered. Service frequencies and offsets are handled analogously to the CSA formulation. RAPTOR computes the earliest arrival times by iterating over a sequence of rounds $k = 0, 1, \ldots, K$, where each round corresponds to routes using at most k transshipments. For each port p, the algorithm maintains a label $\tau_k(p)$, representing the earliest arrival time at port p using at most k transfers. Initialization is given as in Eq. 2.

At each round ($k \geq 1$), RAPTOR processes all services that are reachable from ports improved in the previous round ($k - 1$). For a given service s, the algorithm identifies the earliest boardable port call (p_i, $t^{\text{dep}}{}_i$) such that:

$$\tau_{k-1}(p_i) \leq t_i^{dep} \tag{6}$$

From this point onward, the service is scanned forward, and arrival times at subsequent ports are updated as:

$$\tau_k(p_j) \leftarrow \min\left\{ \tau_k(p_j), t_j^{arr} \right\}, \quad j > i \tag{7}$$

This mechanism ensures that temporal feasibility is strictly respected and that each round captures routes with an additional transfer.

The wait times at ports are evaluated through the comparison between arrival times $\tau_{k-1}(p_i)$ and scheduled departure times t_i^{dep}. A container may board a service only if it arrives at the port sufficiently early, thus ensuring temporal causality without the need for explicit waiting arcs. Transshipment corresponds to transitions between rounds. A route that reaches the destination in k rounds is guaranteed to involve at most k transshipment operations. As a result, RAPTOR provides a framework for controlling and analyzing the trade-off between transit time and the number of transfers. By storing predecessor information in each label update, this approach allows the reconstruction of complete routes, including the sequence of ports and services used. This feature enables the derivation of canonical representations consistent with those obtained from the other modeling approach (CSA), facilitating a direct comparison of routing outcomes.

Conceptually, RAPTOR and CSA both operate on schedule-based representations and avoid the explicit construction of time-expanded networks. However, while CSA focuses on earliest-arrival optimization through a single scan of temporally ordered connections, RAPTOR structures the search around the number of transfers. As a consequence, CSA is particularly efficient for computing single optimal routes, whereas RAPTOR supports multi-criteria extensions and the generation of Pareto optimal solutions. This distinction is relevant in the context of container routing, where routing flexibility and the availability of alternative routes are increasingly important under volatile service deployment conditions.

4 Routing and Comparison Methodology

This section describes the routing procedures adopted for each model and formalizes the methodology used to compare the resulting container routes. Given the structural differences between the representations, a common canonical format is introduced to ensure a consistent and meaningful comparison.

4.1 Routing Procedures

For each modeling approach, routing is performed by solving a minimum-cost route problem between all feasible origin-destination port pairs (o,d) $\in$ P $\times$ P, with o $\neq$ d. All

routing computations are based on generalized time-related costs expressed in hours, ensuring comparability between models. Timetable-based routing algorithms (CSA and RAPTOR) provide a relevant methodological perspective for container routing in liner shipping, and the canonical representation provides a common analytical space in which routing outcomes can be compared across different approaches (Table 1).

Table 1. Conceptual comparison of routing paradigms for liner shipping networks

Feature	CSA	RAPTOR
Network representation	Timetable	Timetable
Temporal feasibility	Explicit	Explicit
Primary optimization focus	Earliest arrival	Transfers (rounds)
Handling of waiting times	Explicit	Explicit
Control of transshipments	Indirect	Explicit
Re-routing capability	High	Moderate
Scalability	High	High

Taken together, the comparison highlights that connection-based and round-based timetable algorithms offer complementary perspectives on container routing in liner shipping networks. While CSA is particularly suited for time-feasible routing and disruption management, round-based approaches such as RAPTOR provide a structured framework to analyze transshipment minimization, which becomes increasingly relevant in highly volatile service environments.

4.2 Canonical Representation

To enable a consistent comparison across the different solutions, each computed route must be transformed into a canonical representation (defined as an ordered sequence of port-service pairs). The canonical sequence stores the ordered succession of ports and their associated liner services, independently of the temporal realization of events or the internal structure of the underlying network model. Service changes between consecutive elements of the sequence explicitly identify transshipment events. The canonical representation is derived from the original routes according to model-specific extraction rules, summarized below. In both approaches, the output consists of an ordered sequence of (temporally) feasible service connections. Each connection directly defines a canonical event through its departure port, arrival port, and associated liner service. Note that waiting times and temporal feasibility are implicitly handled by the ordering of connections and are discarded during canonicalization. The resulting canonical sequence is obtained by collapsing consecutive connections belonging to the same service and retaining only effective service legs.

Figure 1 reports an example of the canonicalization process for CSA and RAPTOR. The generated routes are projected onto an equivalent canonical sequence of

port-service pairs, enabling consistent comparison. Canonical representation retains the essential operational decision (the sequence of ports visited, and the liner services used) while abstracting from model-specific constructs such as diversion nodes, waiting arcs, or connection ordering. Therefore, it defines a unified analytical space in which routing outcomes obtained from different modeling paradigms can be compared in a consistent manner. On this basis, the following section introduces the routing procedures and quantitative metrics used to assess similarities and differences between the routes produced by the two approaches.

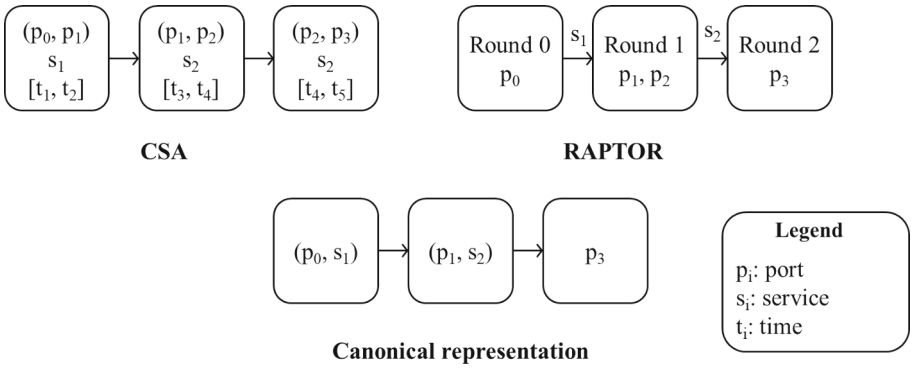

Fig. 1. Routing paradigms and canonical representation

Formally, given a route R between an origin port and a destination port, its canonical representation is defined as follows:

$$R = \left[(p_0, s_1), (p_1, s_2), ..., (p_{k-1}, s_k), p_k\right] \tag{8}$$

where:

- p_0 is the origin port,
- p_k is the destination port,
- $p_i \in P$ are the intermediate ports visited,
- $s_i \in S$ denotes the liner service used to travel from port p_{i-1} to port p_i.

4.3 Similarity Metrics

To quantify the differences between the routing results obtained with different models, several similarity metrics are introduced. All metrics are computed on canonical representations.

Let R^A and R^B be two canonical routes and P^A and P^B the sequences of ports visited. Their similarity is measured using a normalized longest common subsequence (LCS) index:

$$L_p(R^A, R^B) = \frac{LCS(P^A, P^B)}{\max\left(\left|P^A\right|, \left|P^B\right|\right)} \tag{9}$$

The LCS metric is widely used to compare ordered sequences in routing and transportation applications, as it captures structural similarity while allowing for local deviations [33]. Its application to route comparison has been adopted in multimodal routing and network robustness analyses [34].

Similarly, let S^A and S^B denote the corresponding sequences of liner services. Service similarity is defined as:

$$L_s(R^A, R^B) = \frac{LCS(S^A, S^B)}{\max\left(\left|S^A\right|, \left|S^B\right|\right)} \tag{10}$$

This metric focuses on the operational structure of the route, emphasizing the continuity or fragmentation of service usage. Comparing service sequences rather than ports alone is common in public transport and freight network analyses, where service changes represent critical operational events.

The difference in the number of transshipments between two routes is computed as:

$$\Delta T\left(R^A, R^B\right) = T\left(P^A\right) - T\left(P^B\right) \tag{11}$$

The number of transshipments is a key indicator of route quality in liner shipping, as each transshipment entails additional handling, potential delays, and increased operational risk [13, 35]. This metric is therefore commonly used in comparative studies of container routing strategies.

The similarity between the set of route nodes is quantified using the normalized Jaccard index, a standard metric in network analysis to compare set overlap [36, 37]. To capture ordered visitation patterns, the longest common subsequence method is employed, as discussed in related work on route comparison [38, 39]. The node-based Jaccard similarity index measures the overlap between the sets of ports, independently of their order. The Jaccard similarity is defined as:

$$J(R^A, R^B) = \frac{\left|P^A \cap P^B\right|}{\left|P^A \cup P^B\right|} \tag{12}$$

The index takes values in the interval [0, 1], where 1 indicates that the two routes visit exactly the same set of ports. High node-based similarity values indicate agreement on the set of intermediate ports. The LCS metric provides additional information on the order of movements, which is particularly relevant for transshipment routes. Together, these metrics allow for a comparison of different routing solutions.

The comparison analysis is conducted per O/D pair and then aggregated across all considered port pairs. This structured comparison framework allows the identification of systematic differences between the two modeling approaches and provides insight into the conditions under which one representation may be preferred over the other.

5 Application

The objective is to assess the impact of network complexity on routing outcomes and performance under controlled and progressively more complex demanding scenarios. The experiments are carried out in a liner shipping network composed of a set of ports

and regularly scheduled liner services. Each service is defined by an ordered sequence of port calls and associated sailing and port dwell times. A base network configuration is selected as a reference scenario, characterized by a realistic but moderate level of complexity. This base configuration serves as a starting point for generating a family of networks with increasing size and density. All input data are kept identical across the two modeling approaches to ensure comparability of results.

5.1 Scenario Definition

The CMA-CGM [40] shipping lines are considered to define the experimental scenarios. For each service, the ordered sequence of ports calls and indicative sailing times between consecutive calls are available. All services were then converted to a consistent representation using standardized port identifiers and uniform time units; services with incomplete information or ambiguous port sequences were excluded from the analysis. The scenarios of increasing complexity were generated by progressively increasing the number of ports, services, and the length of the planning horizon, as summarized in Table 2. The scenarios are classified into three main categories:

- *Scenario 1* (Small-scale), characterized by a limited number of ports and services, used for qualitative routes comparison and visualization;
- *Scenario 2* (Medium-scale), representative of regional or short-sea shipping networks, used to assess typical routing differences and performance;
- *Scenario 3* (Large-scale), reflecting global liner shipping networks, used to evaluate scalability and potential limitations of the approach.

This classification facilitates the interpretation of results and supports a structured discussion of scalability trade-offs.

Table 2. Classification of experimental scenarios according to network complexity

Scenario ID	Ports	Lines	Services	O/D pairs	Time Horizon [days]
1	19	4	4	342	42
2	47	6	6	2162	105
3	416	310	313	172,640	161

Although this approach allows controlled experimentation, it may not fully capture all sources of uncertainty observed in real-world operations. The use of a public dataset ensures transparency and facilitates the reproducibility of the experimentation.

5.2 Results

Table 3 and Table 4 report a summary of the results. CSA processes a chronologically ordered set of connections, and the number of connections grows with the complexity of the scenario (about 122,000 connections in scenario 3). Preparation times are low (always

below 0.1s except in scenario 3), reflecting the lightweight nature of connection generation. However, scan times increase sharply in scenario 3, where the exhaustive sweep of all connections leads to a computational burden (exceeding 8,600 s). This behavior highlights how CSA maintains memory efficiency but may become time consuming in extreme-scale networks with dense service schedules.

RAPTOR exhibits similar behavior. Calendar generation remains inexpensive, staying under 1.5 s even in scenario 3, confirming the efficiency of its preprocessing stage. Scan times, however, grow with input size: from around 2 s in scenario 1 to over 856,000 s in scenario 3. This underscores RAPTOR's sensitivity to the number of stops and services when applied to dense, large liner networks.

Table 3. CSA: connections and computational effort

	Scenario 1	Scenario 2	Scenario 3
Connections	1,089	4,452	122,088
Prep. Time [s]	0.031	0.046	3.374
Scan Time [s]	0.055	2.740	8,635.360

CSA offers a graph-free, temporally consistent alternative with low memory requirements, though scan times may become prohibitive in very large cases. RAPTOR remains competitive in small and medium instances but encounters substantial performance degradation as network density increases. These contrasts have implications for selecting routing methodologies in increasingly large-scale liner shipping networks.

Table 4. RAPTOR: network size and computational effort for the routing approaches

	Scenario 1	Scenario 2	Scenario 3
Calendar Gen. Time [s]	0.057	0.090	1.464
Scan Time [s]	2.022	310.17	856,387.92

Table 5 summarizes the main routing indicators averaging over all origin-destination pairs for the two procedures in the three scenarios.

Excluding scenario 1, the CSA approach generates longer travel times than RAPTOR. This trend is accompanied by an increase in the number of transshipments: in CSA, the average number and the maximum number of transshipments (from 5.35 to 7.53 and from 2 to 15 respectively) are greater than in RAPTOR (from 2.49 to 4.57 and from 1 to 9 respectively).

The route metrics, based on canonical representation, era reported in Table 6, allowing a model-independent comparison of routing outcomes.

Pairwise comparisons between CSA and RAP confirm that these two temporally oriented algorithms diverge substantially in complex settings. While similarity levels

Table 5. Routing outcomes averaged over all origin-destination pairs

Scenario	Procedure	X_1(*)	X_2(*)	X_3(*)	X_4(*)
Scenario 1	CSA	2760.80	5.35	0.91	2
	RAPTOR	2771.50	2.49	0.49	1
Scenario 2	CSA	2194.90	7.78	2.21	5
	RAPTOR	1789.90	3.15	1.15	3
Scenario 3	CSA	1495.70	7.53	5.56	15
	RAPTOR	1435.80	4.57	2.57	9

(*)X_1, average time in hours; X_2, average number of ports; X_3, average number of transshipments; X_4, maximum number of transshipments.

remain moderate in scenario 1 (Jaccard 0.80), they drop in scenarios 2 and 3, with identical routes falling to 12.30% in scenario 2, and below 1% in scenario 3.

These findings underscore the importance of canonical representations, making such comparisons possible despite the fundamentally different modeling paradigms underlying each routing method.

The similarity analysis lacks in pointing out the overall trade-offs between modeling paradigms when network size, routing outcomes, computational effort, and structural divergence are jointly considered. To synthesize these heterogeneous dimensions, a a weighted average synthesis of the global similarity patterns is reported in Table 6.

Table 6. Similarity metrics based on canonical representation

Scenario	Jaccard	LCS	Identical [%]
Scenario 1	0.80	0.82	16.70
Scenario 2	0.55	0.62	12.30
Scenario 3	0.47	0.53	0.90

6 Discussion and Conclusions

The liner shipping industry is undergoing a profound transformation, marked by increasing operational volatility and frequent reconfiguration of service networks. Recent evidence on record high-capacity churn across major trade lanes confirms that the relative stability characterizing pre-pandemic liner networks has largely disappeared. In this context, the choice of routing model is no longer a purely technical issue, but a substantive modeling decision with direct implications for how network behavior, resilience, and feasibility are represented.

The pairwise comparison between the Connection Scan Algorithm and RAPTOR highlights how their performance profiles respond to varying network conditions, offering valuable insights to support decision-making in liner shipping. Using canonical

representation (that enables model-independent analysis of routing results), this study supports the selection of the most suitable routing algorithm to minimize transit times, reduce the number of transfers, and manage computational resources.

The CSA produces routing outcomes that diverge from RAPTOR, with similarity scores dropping as network complexity increases. CSA routes are longer in terms of travel time, exhibit the highest numbers of transshipments, and differ the most in terms of service and port sequences. Identical-route percentages fall to nearly zero in the largest scenarios for all comparisons. This pattern reflects the algorithm's temporal, earliest-arrival logic: CSA exploits every feasible connection in chronological order, exploring a far richer and more fragmented set of temporal opportunities. As a result, CSA uncovers routes that remain inaccessible to round-based temporal scans (RAPTOR). In fact, RAPTOR produces spatially compact routes, visiting substantially fewer ports. Its multi-round scanning logic prioritizes temporally efficient transfer sequences.

From a modeling standpoint, these findings highlight a spectrum of trade-offs between structural stability and temporal responsiveness:

- CSA emerges as a powerful tool for disruption-aware and operational-level decision making, where fine-grained temporal changes, missed connections, or irregular schedules require rapid re-evaluation of earliest-arrival opportunities;
- RAPTOR offers a balance between temporal realism and spatial compactness, making it suitable for medium-scale operational analyses where fast computation and moderate temporal accuracy are required.

The methodologies and comparative insights presented in this study are of direct interest to several stakeholders. Shipping line operators and network planners can leverage the findings to select routing strategies for minimizing transshipments or improving schedule robustness. Port authorities and terminal managers may take advantage of the approach for managing incoming containers flows. In addition, developers of maritime decision-support systems and logistics technology providers may apply these results to enhance the design of routing engines. Finally, policy makers may find comparative analysis valuable for scenario assessment and long-term strategy development.

Even though this study offers a comprehensive comparison of routing algorithms under controlled scenarios, several limitations should be marked. The analysis is based on deterministic routing, without considering elements such as unexpected delays or schedule reliability. Besides, the procedures may require additional adaptation to specific IT or integration with other systems. Addressing these limitations opens opportunities for future research, including real-time data integration and the exploration of hybrid or probabilistic routing models.

This work raises several perspectives for future research. A first direction concerns hybrid modeling, combining structural stability from line based or time expanded network models with temporal reactivity from CSA or RAPTOR, for example through multi-stage routing or selective temporal expansion. A second direction involves multi-objective formulations, incorporating reliability, buffer times, or service regularity alongside earliest-arrival logic to mitigate the excessive fragmentation observed in CSA-based solutions. Further developments could also explore transfer-penalized connection scanning, priority-aware RAPTOR variants, or probabilistic schedule models to better represent uncertainty.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Russo, F., Musolino, G., Assumma, V.: An integrated procedure to estimate demand flows of maritime container transport at international scale. Int. J. Shipp. Transp. Logist. **6**, 112–132 (2014). https://doi.org/10.1504/IJSTL.2014.059566
2. Russo, F., Pedà, G., Musolino, G.: Attributes influencing port times of container ships. WMU J. Marit. Affairs **23**, 375–392 (2024). https://doi.org/10.1007/s13437-024-00336-7
3. Russo, F., Musolino, G.: Transportation system models to analyse ports competition and cooperation. WMU J. Marit. Affairs **23**, 393–413 (2024). https://doi.org/10.1007/s13437-024-00345-6
4. Musolino, G., Cartisano, A., Fortugno, G.: Structural factors for a third-generation port: planning interventions for mechanical logistics in Gioia Tauro, Italy. WIT Trans. Built Environ. **204**, 55–66 (2021). https://doi.org/10.2495/UT210051
5. Wang, S., Meng, Q.: Schedule design and container routing in liner shipping. Transp. Res. Rec. **2222**, 25–33 (2011). https://doi.org/10.3141/2222-04
6. Wang, S., Meng, Q., Sun, Z.: Container routing in liner shipping. Transp. Res. Part E Logist. Transp. Rev. **49**, 1–7 (2013). https://doi.org/10.1016/j.tre.2012.06.009
7. Baird, A.J.: The economics of container transhipment in northern Europe. Marit. Econ. Logist. **4**, 249–280 (2002). https://doi.org/10.1057/palgrave.ijme.9100048
8. Baird, A.J.: Optimising the container transhipment hub location in northern Europe. J. Transp. Geogr. **14**, 195–214 (2006). https://doi.org/10.1016/j.jtrangeo.2004.12.004
9. Kozan, E., Preston, P.: Genetic algorithms to schedule container transfers at multimodal terminals. Int. Trans. Oper. Res. **6**, 311–329 (1999). https://doi.org/10.1111/j.1475-3995.1999.tb00158.x
10. Alharbi, A., Wang, S., Davy, P.: Schedule design for sustainable container supply chain networks with port time windows. Adv. Eng. Inform. **29**, 322–331 (2015). https://doi.org/10.1016/j.aei.2014.12.001
11. Sea Intelligence: Sea-Intelligence - Press Room. https://www.sea-intelligence.com/press-room/. Accessed 02 Mar 2026
12. Delling, D., Pajor, T., Werneck, R.F.: Round-based public transit routing. Transp. Sci. **49**, 591–604 (2015). https://doi.org/10.1287/trsc.2014.0534
13. Balakrishnan, A., Karsten, C.V.: Container shipping service selection and cargo routing with transshipment limits. Eur. J. Oper. Res. **263**, 652–663 (2017). https://doi.org/10.1016/j.ejor.2017.05.031
14. Castilla-Rodríguez, I., Expósito-Izquierdo, C., Melián-Batista, B., Aguilar, R.M., Moreno-Vega, J.M.: Simulation-optimization for the management of the transshipment operations at maritime container terminals. Expert Syst. Appl. **139**, 112852 (2020). https://doi.org/10.1016/j.eswa.2019.112852
15. Notteboom, T.E.: The time factor in liner shipping services. Marit. Econ. Logist. **8**, 19–39 (2006). https://doi.org/10.1057/palgrave.mel.9100148
16. Wang, T., Wang, S., Meng, Q.: Liner Ship Fleet Planning: Models and Algorithms. Elsevier, Amsterdam (2017)
17. Zheng, J., Hou, X., Qi, J., Yang, L.: Liner ship scheduling with time-dependent port charges. Marit. Policy Manag. **49**, 18–38 (2022). https://doi.org/10.1080/03088839.2020.1849840
18. Dibbelt, J., Pajor, T., Strasser, B., Wagner, D.: Connection scan algorithm. ACM J. Exp. Algorithmics **23**, 1.7:1–1.7:56 (2018). https://doi.org/10.1145/3274661

19. Pan, J.-J., Zhang, Y.-F., Fan, B.: Strengthening container shipping network connectivity during COVID-19: a graph theory approach. Ocean Coast. Manag. **229**, 106338 (2022). https://doi.org/10.1016/j.ocecoaman.2022.106338
20. Jiang, L., Chen, L., Wang, W., Wei, W., Lv, Z., Wang, H.: Advanced network representation learning for container shipping network analysis. IEEE Netw. **35**, 182–187 (2021). https://doi.org/10.1109/MNET.011.2000444
21. Huang, C.: The optimization of empty shipping container reposition based on the network simplex method. In: 2011 International Conference on Management and Service Science, pp. 1–4 (2011). https://doi.org/10.1109/ICMSS.2011.5998249
22. Martin, F., Hemmelmayr, V.C., Wakolbinger, T.: Integrated express shipment service network design with customer choice and endogenous delivery time restrictions. Eur. J. Oper. Res. **294**, 590–603 (2021). https://doi.org/10.1016/j.ejor.2021.02.014
23. Gallo, G., Pallottino, S.: Shortest path methods: a unifying approach. In: Gallo, G., Sandi, C. (eds.) Netflow at Pisa. Mathematical Programming Studies, vol. 26, pp. 38–64. Springer, Heidelberg (1986). https://doi.org/10.1007/BFb0121087
24. Spiess, H., Florian, M.: Optimal strategies: a new assignment model for transit networks. Transp. Res. Part B Methodol. **23**, 83–102 (1989). https://doi.org/10.1016/0191-2615(89)90034-9
25. Nguyen, S., Pallottino, S.: Equilibrium traffic assignment for large scale transit networks. Eur. J. Oper. Res. **37**, 176–186 (1988). https://doi.org/10.1016/0377-2217(88)90327-X
26. Di Gangi, M., Cantarella, G.E., Vitetta, A.: Solving stochastic frequency-based assignment to transit networks with pre-trip/en-route path choice. EURO J. Transp. Logist. **8**, 661–681 (2019). https://doi.org/10.1007/s13676-019-00142-9
27. Nguyen, S., Pallottino, S., Gendreau, M.: Implicit enumeration of hyperpaths in a logit model for transit networks. Transp. Sci. **32**, 54–64 (1998). https://doi.org/10.1287/trsc.32.1.54
28. Nuzzolo, A., Comi, A.: A normative optimal strategy in intelligent transit networks. Transp. Res. Procedia **27**, 380–387 (2017). https://doi.org/10.1016/j.trpro.2017.12.088
29. Cooke, K.L., Halsey, E.: The shortest route through a network with time-dependent internodal transit times. J. Math. Anal. Appl. **14**, 493–498 (1966). https://doi.org/10.1016/0022-247X(66)90009-6
30. Liu, T., (Avi) Ceder, A.: Integrated public transport timetable synchronization and vehicle scheduling with demand assignment: a bi-objective bi-level model using deficit function approach. Transp. Res. Procedia **23**, 341–361 (2017). https://doi.org/10.1016/j.trpro.2017.05.020
31. Liu, T., (Avi) Ceder, A.: Research in public transport vehicle scheduling. In: Handbook of Public Transport Research, pp. 388–408. Edward Elgar Publishing (2021)
32. van Lieshout, R.N., van der Schaft, T.: Dynamic discretization discovery for the multidepot vehicle scheduling problem with trip shifting. INFORMS J. Comput. (2025). https://doi.org/10.1287/ijoc.2024.0698
33. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. McGraw-Hill (2009)
34. Huang, D., Liu, Z., Jia, Z., Liu, Z.: Customised bus routing problem for commuting travel demand considering route similarity. Transportmetrica A Transp. Sci., 1–28 (2025). https://doi.org/10.1080/23249935.2025.2528960
35. Christiansen, M., Fagerholt, K., Nygreen, B., Ronen, D.: Ship routing and scheduling in the new millennium. Eur. J. Oper. Res. **228**, 467–483 (2013). https://doi.org/10.1016/j.ejor.2012.12.002
36. Jaccard, P.: Nouvelles recherches sur la distribution florale. Bulletin de la Société vaudoise des Sciences Naturelles **44**, 223–270 (1908)

37. Hwang, C.-M., Yang, M.-S., Hung, W.-L.: New similarity measures of intuitionistic fuzzy sets based on the Jaccard index with its application to clustering. Int. J. Intell. Syst. **33**, 1672–1688 (2018). https://doi.org/10.1002/int.21990
38. Chen, L., Özsu, M.T., Oria, V.: Robust and fast similarity search for moving object trajectories. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, New York, NY, USA, pp. 491–502. Association for Computing Machinery (2005). https://doi.org/10.1145/1066157.1066213
39. Vlachos, M., Kollios, G., Gunopulos, D.: Discovering similar multidimensional trajectories. In: Proceedings 18th International Conference on Data Engineering, pp. 673–684 (2002). https://doi.org/10.1109/ICDE.2002.994784
40. CMA CGM: CMA CGM | Search shipping lines. https://www.cma-cgm.com. Accessed 13 Jan 2025