DTU Library

The multi-commodity network flow problem with soft transit time constraints: Application to liner shipping

Trivella, Alessio; Corman, Francesco; Koza, David F.; Pisinger, David

Published in:
Transportation Research Part E: Logistics and Transportation Review

Link to article, DOI:
10.1016/j.tre.2021.102342

Publication date:
2021

Document Version
Publisher's PDF, also known as Version of record

Transportation Research Part E 150 (2021) 102342

Contents lists available at ScienceDirect

Transportation Research Part E

journal homepage: www.elsevier.com/locate/tre

The multi-commodity network flow problem with soft transit time constraints: Application to liner shipping

Alessio Trivella [a,*], Francesco Corman [a], David F. Koza [b], David Pisinger [c]

[a] Institute for Transport Planning and Systems, ETH Zürich, 8093 Zürich, Switzerland
[b] Vattenfall Vindkraft A/S, 6000 Kolding, Denmark
[c] DTU Management, Technical University of Denmark, 2800 Kongens Lyngby, Denmark

ARTICLE INFO

Keywords:
Networks
Multi-commodity flow
Column generation
Transit time
Liner shipping

ABSTRACT

The multi-commodity network flow problem (MCNF) consists in routing a set of commodities through a capacitated network at minimum cost and is relevant for routing containers in liner shipping networks. As commodity transit times are often a critical factor, the literature has introduced hard limits on commodity transit times. In practical contexts, however, these hard limits may fail to provide sufficient flexibility since routes with even tiny delays would be discarded. Motivated by a major liner shipping operator, we study an MCNF generalization where transit time restrictions are modeled as soft constraints, in which delays are discouraged using penalty functions of transit time. Similarly, early commodity arrivals can receive a discount in cost. We derive properties that distinguish this model from other MCNF variants and adapt a column generation procedure to efficiently solve it. Extensive numerical experiments conducted on realistic liner shipping instances reveal that the explicit consideration of penalty functions can lead to significant cost reductions compared to hard transit time deadlines. Moreover, the penalties can be used to steer the flow towards slower or faster configurations, resulting in a potential increase in operational costs, which generates a trade-off that we quantify under varying penalty functions.

1. Introduction

The multi-commodity network flow problem (MCNF) consists in routing a set of commodities through a capacitated network, from their respective origins to demand destinations, so that the capacity limits on the network arcs are respected and the overall commodity transportation cost is minimized. The MCNF is well known in the operations research community and has been studied for decades (see e.g. Ahuja et al., 1993, Ch.17, for a good introduction). The MCNF has practical relevance in applications spanning from transportation, e.g., in efficiently routing cargo over railway or liner shipping networks (Agarwal and Ergun, 2008; Wang et al., 2013b; Karsten et al., 2015; Azadi Moghaddam Arani et al., 2019; Christiansen et al., 2019; Koza, 2019) to telecommunication and computer applications where commodities represent communication pairs (Ahuja et al., 1993; Barnhart et al., 2000; Holmberg and Yuan, 2003; Minoux, 2006) including Internet and satellite networks (Li et al., 2017; Zhang et al., 2018; Wang et al., 2019). Even though the MCNF can be formulated and solved as a linear program, it is known that this formulation may be intractable for very large networks such as realistic liner shipping networks with a large number of commodities to ship (Brouer et al., 2011). Rather, the

* Corresponding author.
E-mail addresses: alessio.trivella@ivt.baug.ethz.ch (A. Trivella), francesco.corman@ivt.baug.ethz.ch (F. Corman), davidfranz.koza@vattenfall.com (D.F. Koza), dapi@dtu.dk (D. Pisinger).

https://doi.org/10.1016/j.tre.2021.102342
Received 11 December 2020; Received in revised form 7 March 2021; Accepted 17 April 2021
Available online 23 April 2021

MCNF is usually solved much more efficiently using a path-flow formulation and a delayed column generation (CG) method (Ahuja et al., 1993; Desaulniers et al., 2005).

The standard MCNF does not consider transit times of commodities in the network and the only objective is to minimize the total cost of the flow under network capacity constraints. In the liner shipping industry, however, the commodity transit time represents a critical factor that needs to be controlled, for instance to ensure a competitive service level to customers. Indeed, cost and time are usually identified in the literature as the most important factors in applications involving shipping services (Notteboom, 2006; Gelareh et al., 2010; Brouer et al., 2013a; Meng et al., 2013; Karsten et al., 2015). Therefore, the *hard transit time-constrained* MCNF (henceforth HTC-MCNF) has been proposed in the literature to account for commodity transit times (Holmberg and Yuan, 2003; Wang and Meng, 2014; Karsten et al., 2015; Koza et al., 2020). Specifically, these studies incorporate transit time into the MCNF as an explicit constraint, i.e., by imposing a maximum allowed transit time for each commodity. The HTC-MCNF can be solved by CG, where the master problem is essentially a set covering model (as it is in the case of the standard MCNF) and the subproblem is a resource-constrained shortest path problem (RCSP) for each commodity that is solved using either a label-setting algorithm (Karsten et al., 2015; Koza et al., 2020) or a label-correcting algorithm (Holmberg and Yuan, 2003) based on dynamic programming (Irnich and Desaulniers, 2005).

1.1. Need for soft transit time constraints

Including hard limits on commodity transit times is one way to restrict time in the MCNF, resulting in commodity flow solutions that try to avoid routes of long duration. Despite being academically relevant, this approach may be too inflexible to be adopted in practical contexts since cargo flow solutions that exceed the target transit time would be discarded, even when the delay is tiny. In practice, simply defining an economically reasonable transit time limit per market or even per customer within a market is a non-trivial task. On an operational level, a liner shipping operator may indeed accept longer transit times for selected cargo routes if the overall cargo flow benefits. A second inherent assumption of the HTC-MCNF is that two cargo paths between the same origin–destination (O–D) pair are equivalent independent of their transit time, as long as their transit times are below the hard limit; this assumption, however, does not hold in practice. The transit time is particularly relevant for time sensitive goods such as perishable goods or goods of high value (Notteboom, 2006; Wang and Meng, 2014). Some shippers are indeed willing to pay a premium for having their cargo shipped at a shorter transit time, while for others, transit time is of lesser importance (Othelius and Wemmert, 2014). So-called express services in liner shipping commonly offer faster transit times at higher rates and hence exploit different transit time requirements among shippers.

Our criticism towards the HTC-MCNF model emerged after discussions with a major liner shipping operator, which would rather discourage routes with small delays but would accept these routes if the overall cargo flow would benefit from it. These considerations motivated the research in this paper. Ideally, we do not want to exclude a priori longer routes but instead punish them using a penalty which is a function of the delay, i.e., the extra time taken to route a commodity with respect to the target transit time established for that commodity. The larger the delay, the more costly the route will be, reflecting e.g. increased inventory costs or decreased customer satisfaction. The goal of this paper is to close this gap in the literature by proposing and solving the *soft transit time-constrained* MCNF (in the following STC-MCNF) that uses generic non-decreasing *penalty functions* to punish violations of the target transit time and to encourage the use of faster routes. The STC-MCNF introduced in this work allows to explicitly model the trade-off between transit time and related transportation cost on a trade, commodity or even customer level. It can be used to solve day-to-day operational cargo routing problems, but we also show that it could replace the less accurate HTC-MCNF within network design solution algorithms without increasing runtimes.

1.2. Overview of the work

We start by presenting some properties of the STC-MCNF and showing how it relates to other problems. We first show that the HTC-MCNF can be modeled as a special case of the STC-MCNF. We then provide theoretical support for studying the STC-MCNF by constructing sequences of instances that exhibit increasing (unbounded) savings from employing the STC-MCNF compared to (i) the HTC-MCNF, and (ii) a sequential approach that solves a standard MCNF first and subsequently applies the penalties. Moreover, we show that it is not possible in general to solve the STC-MCNF by simply modifying the arc costs in the graph to include the penalties and solving a standard MCNF in this modified graph. We then discuss the complexity of the STC-MCNF.

To solve real-life instances of the problem, we extended the standard MCNF path-flow formulation to the STC-MCNF. This can be done by incorporating the penalty term directly into the cost of each route, meaning that the formulation is similar to the standard MCNF but with updated route cost parameters. Nevertheless, adapting CG to solve this formulation is non-trivial because the subproblem requires determining for each commodity the path that minimizes the sum of reduced arc costs and the delay penalty. We formulate this subproblem as an RCSP variant that is solved using a label-correcting algorithm. To keep the algorithm tractable in the presence of soft transit time constraints, we enforce different fathoming criteria, including adaptations to our setting of goal-based fathoming rules (Reinhardt and Pisinger, 2011).

We perform extensive computational experiments using realistic liner shipping networks from LINER-LIB (Brouer et al., 2012, 2013a). We test our column generation approach for the STC-MCNF on a total of 60 networks combined with different penalty functions used in the liner shipping literature (Reinhardt et al., 2020). We found that our CG method could solve to optimality all instances in a few seconds (our largest instances contain about 370 nodes, 4600 arcs and 2800 commodities), hence emerging as a very efficient method for solving the STC-MCNF. For comparison, we also implement the HTC-MCNF solution and the simpler sequential heuristic where a standard MCNF is solved first and the penalties are applied afterwards. We show that the cost incurred under both approaches can be substantially larger than the optimal STC-MCNF cost, highlighting the importance of considering soft transit time constraints explicitly. We also found that penalty functions can be used to steer the flow towards slower or faster configurations, at a potential increase in operational costs. This generates a trade-off between operating costs and the average flow transit time, which the liner shipping operator should evaluate and that we help quantifying under varying penalty functions. Our results inform decision making in liner shipping by examining the effect of penalty functions and their parameterization on the container flow, its cost, and transit time.

1.3. Paper structure

We review the related literature and summarize our contributions in Section 2. In Section 3, we formalize the penalty functions and discuss the STC-MCNF properties. In Section 4, we explain how to extend the MCNF path-flow formulation to account for soft transit time constraints. We present our liner shipping case study in Section 5 and the numerical experiments in Section 6. We conclude in Section 7. Additional models and experiments are reported in an online supplement.

2. Novelty and related works

Due to the vast amount of literature on the MCNF and related problems, e.g. the liner shipping network design problem, in Section 2.1 we review the studies that explicitly consider the transit time of commodities in some way. We also focus on (but do not restrict to) liner shipping applications, from which most of the available literature relevant to the STC-MCNF has originated. We then summarize our contributions in Section 2.2.

2.1. Literature review

First, highly relevant to our STC-MCNF are the papers already discussed dealing with the HTC-MCNF (Holmberg and Yuan, 2003; Karsten et al., 2015; Koza et al., 2020). Worthwhile reviewing are then time extensions to the liner shipping network design problem, of which the MCNF is in fact a subproblem. In addition to Wang and Meng (2014) and Koza et al. (2020) dealing with a network design with transit time deadlines, Koza (2017)[Ch.3] discusses how to incorporate time-dependent revenues in a network design problem, while Fragkos et al. (2017) study a multi-period network design problem in which arcs can be activated in different periods. Several other studies in the liner shipping domain have accounted for commodity transit times while optimizing the speed of vessels (Brouer et al., 2013b; Venturini et al., 2017; Reinhardt et al., 2020). Brouer et al. (2013b) and Reinhardt et al. (2020) use penalty functions in the objective to penalize longer and reward shorter transit times in the same way as we do. However, dealing with a speed optimization problem, the trade-off investigated in these papers is between delay costs and fuel consumption on sailing legs rather that cargo routing decisions. The joint cargo allocation and speed optimization problem is tackled including either hard transit time constraints (Guericke and Tierney, 2015) or minimum steaming speed of vessels (Xia et al., 2015), where both approaches are motivated by the need for liner shipping operators to provide customers with competitive service levels.

Two works consider a time-dependent commodity demand (Wang et al., 2013a, 2016), which is similar in spirit to the penalty function that we use, but not equivalent. Indeed, demand depreciates over time and time-dependency shows up in the constraints, i.e. only a proportion of this demand has to be fulfilled depending on transit time (in our problem all demand must be fulfilled, potentially penalized). In particular, Wang et al. (2013a) tackle a scheduling problem in which the route is given as input, i.e. there is no commodity routing decision which is instead the problem investigated in our paper. Wang et al. (2016) study a liner assignment problem which resembles our STC-MCNF in several aspects. However, they: (1) use time-sensitive demands while we use penalty functions in the objective, (2) define a model that is only applicable to liner shipping (e.g. by using liner shipping-specific constraints), while our problem definition and solution methodology are more general (although motivated by and tested on a liner shipping application), (3) do not provide a path-flow formulation solved with CG but only an arc-flow formulation, which is known to be intractable for large instances, and (4) test time-sensitive demands only on a toy example with two ports while experiments conducted on larger networks use instead hard transit time limits. In contrast, testing the effect of soft transit time constraints under different penalty functions is the focus of our numerical study.

Other works consider commodity flow problems in *delay tolerant networks*, e.g. satellite and Internet networks, in which capacity on arcs is time dependent (see, Li et al., 2017; Zhang et al., 2018, and references therein). In general, commodity flow problems on delay tolerant networks are quite different compared to the problem studied here since arc capacities in the STC-MCNF do not vary over time. Finally, a recent publication deals with the "quickest" MCNF (Melchiori and Sgalambro, 2020) where the goal is to minimize the total makespan, i.e. the time needed for all commodities to be routed. Despite being an MCNF variant involving transit times, this problem does not consider the trade-off between cost and time and is hence different in nature from the STC-MCNF. Possible applications where instead STC-MCNF could be applied are discussed in Online Supplement E.

Our solution approach is based on CG where the pricing problem is an RCSP variant solved with dynamic programming. This approach is similar to the one used by Holmberg and Yuan (2003), Karsten et al. (2015), and Koza et al. (2020) for the HTC-MCNF. However, in our setting, it is not possible to delete labels (i.e., partial shortest path solutions) that exceed the hard transit time limits, which was a key feature in those papers, to reduce the number of labels. Instead, we adapted state-of-the-art fathoming rules to solve the pricing problem exactly (Reinhardt and Pisinger, 2011). Due to favorable running time results we did not explore other strategies, but mention here that several techniques exist that could potentially speed up this process further, including alternative exact methods or heuristic pricing schemes (Boland et al., 2006; Lozano and Medaglia, 2013; Pugliese and Guerriero, 2013; Duque et al., 2015; Sedeño-Noda and Colebrook, 2019; Enzi et al., 2021).

2.2. Summary of contributions

The contributions of this paper are summarized in the following four items:

1. Motivated by an industry collaboration, we introduce for the first time the STC-MCNF, which is relevant in practice for flowing cargo in liner shipping networks and significantly generalizes the existing HTC-MCNF.
2. We provide theoretical insights on the STC-MCNF that help understanding how this problem relates to both the HTC-MCNF and the standard MCNF.
3. We present a path-flow formulation of the STC-MCNF and a very efficient CG method to solve it to optimality, by using and adapting state-of-the-art fathoming rules in the subproblem. We also derive an arc-flow model formulation of STC-MCNF, but relegate this model and related experiments to Online Supplement B due to inferior numerical performance.
4. We perform extensive numerical experiments on realistic liner shipping networks. We discuss economic effects, trade-offs, and the impact of the penalty function and its parameters on shipping cost and transit time. Our insights help inform cargo routing decisions in liner shipping.

3. Multi-commodity flow with soft transit time constraints

In this section, we introduce soft transit time constraints for multi-commodity network flows. The STC-MCNF can be defined as follows: we are given a directed weighted graph $\mathcal{G} = (\mathcal{N}, \mathcal{A}, c, u, t)$, where $\mathcal{N}$ and $\mathcal{A}$ are the sets of nodes and arcs (i.e. directed edges) of the graph, and for each arc $(i, j) \in \mathcal{A}$, $c_{ij} \geqslant 0$, $u_{ij} \geqslant 0$, and $t_{ij} \geqslant 0$ denote respectively the unit traversal cost, the capacity of the arc, and the transit time associated with the arc. Moreover, we are given K commodities belonging to set $\mathcal{K} = \{1, \dots, K\}$, where each commodity k is defined by an origin (or source) $o_k \in \mathcal{N}$, a destination (or sink) $s_k \in \mathcal{N}$, $s_k \neq o_k$, a demand $d_k > 0$, and a delay penalty function $f_k(t)$ of the transit time duration t. The total transit time duration of a feasible origin–destination path of a commodity consists of the sum of transit times t_{ij} of all arcs used by the path. The objective of the STC-MCNF is to ship all commodities from their origin to their destination, respecting arc capacities and minimizing the sum of arc traversal costs and delay penalties.

In Section 3.1 we formally define the penalty functions. In Section 3.2, we present an illustrative example comparing MCNF, HTC-MCNF, and STC-MCNF on a small network. In Section 3.3, we highlight some properties that distinguish these problems and further characterize the STC-MCNF.

3.1. Penalty functions of transit time

In the HTC-MCNF, paths for a commodity k exceeding a maximum allowed transit time t_k^{MAX} are deemed not acceptable (infeasible), and hence discarded (Karsten et al., 2015). To model the impact of transit time in a more flexible manner, we introduce for each commodity k a penalty function $f_k : \mathbb{R}_+ \mapsto \mathbb{R}$ which penalizes delays with respect to the target transit time t_k^{MAX}. The idea of using such function is to avoid discarding a priori the routes that exceed the transit time t_k^{MAX} but allowing the model to choose such solutions at a higher (penalized) cost. Indeed, this flexibility could benefit the overall flow solution by penalizing one commodity while enabling other commodities to take faster and/or cheaper routes. Additionally, the function f_k can also be used to provide a discount to those paths requiring a transit time shorter than t_k^{MAX}, that is, to encourage choosing faster routes. The penalty function is defined per unit of commodity flow. Notice, that two or more commodities may share the same origin–destination pair but have different penalty functions to describe, for example, demand of different priority between the same pair of ports/cities.

We provide three examples of penalty functions for a single commodity in Fig. 1. Fig. 1(a) displays a simple piecewise constant function that is zero in the interval $[0, t_k^{\mathrm{MAX}}]$ and equal to a positive constant for $t > t_k^{\mathrm{MAX}}$. This penalty was used e.g. by Brouer et al. (2013b). Fig. 1(b) shows instead a piecewise linear penalty function which is zero at $t = t_k^{\mathrm{MAX}}$ and linearly increasing for larger t values with a slope $\alpha_k \geqslant 0$. Moreover, due to a linear discount with a second slope $\beta_k \geqslant 0$ (typically $\beta_k < \alpha_k$), this function favors

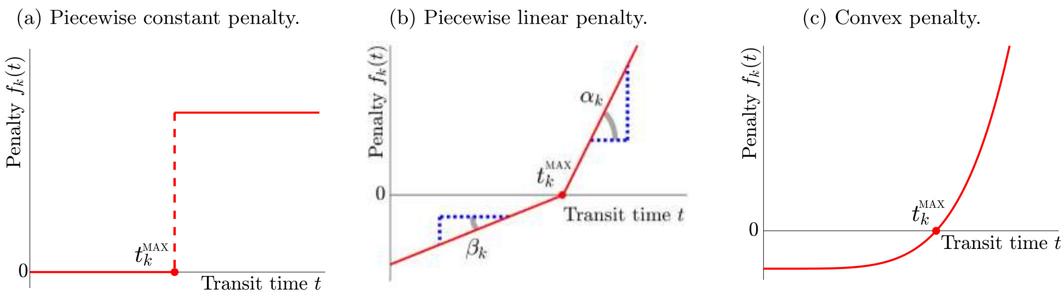


Fig. 1. Examples of penalty function $f_k(t)$ for a commodity k.

the use of faster routes even when the target transit time t_k^{MAX} is met. This penalty was used by Wang and Meng (2012) (without discounts) and by Reinhardt et al. (2020) (with discounts) and is formally expressed by:

$$f_k(t) := \begin{cases} \beta_k(t - t_k^{\text{MAX}}) & \text{for } t \in [0, t_k^{\text{MAX}}], \\ \alpha_k(t - t_k^{\text{MAX}}) & \text{for } t > t_k^{\text{MAX}}. \end{cases} \tag{1}$$

Finally, Fig. 1(c) shows a convex penalty function that also includes an early arrival discount, which was used in Pisinger (2016). Choosing such a penalty could be motivated by the intuition that for some applications exceeding t_k^{MAX} might result in costs of higher marginal increments for each extra unit of delay.

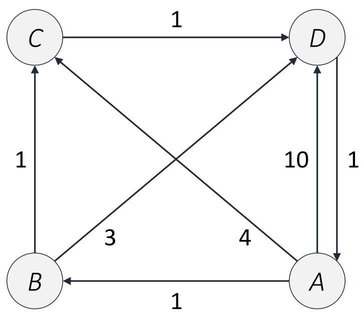
The methodology we develop in Section 4 does not require convexity or other functional properties and can handle functions more general than those shown in Fig. 1. We will only assume that penalties are non-decreasing functions, which seems a mild and reasonable assumption as there is no gain for commodities to take longer transit times, at least in the considered application.

3.2. An illustrative example

To clarify our problem setting, we present an illustrative example on a small-scale instance described in Fig. 2. The network that we use in this example is composed by 4 nodes $\mathcal{N} = \{A, B, C, D\}$ and 7 arcs. The values reported on the arcs represent the arc unit costs c_{ij}. The arc capacity u_{ij} and transit time t_{ij} are the same for all arcs and are equal to 3 and 1, respectively (we omit these values in the figure). Two commodities must be sent through the network. For each commodity k, the figure reports its origin o_k, destination s_k, demand d_k, and three parameters that uniquely specify a piecewise linear penalty function as the one previously shown in Fig. 1(b) and Eq. (1). These parameters are the maximum transit time for which no penalty is applied (t_k^{MAX}) and the two slopes for delays (α_k) and early arrivals (β_k), the latter of which is set to zero in this example.

We start by solving the standard MCNF, that is, commodity transit time is neglected for now. The solution for this case is reported in Fig. 3(a) and the total flow cost is 13 ($3 \cdot 2 = 6$ for the first commodity and $3 + 4 = 7$ for the second). Clearly, commodities tend to follow the cheapest paths that use the arcs with cost equal to 1. Due to capacity constraints, one unit of the second commodity takes a slightly more expensive path. The first commodity needs 3 units of time to reach destination while the second takes partly 2 and partly 3 units of time, depending on the path.

Next, we consider the HTC-MCNF, where limits on transit time are set to the values t_k^{MAX} from Fig. 2. We report the solution in Fig. 3(b). It is easy to see that the set of paths fulfilling the limits is restricted to a single path for each commodities, namely $\{AD\}$ for commodity 1 and $\{BDA\}$ for commodity 2. All other paths violate the limits. The commodities are therefore obliged to follow these paths, which represent the only feasible solution to the problem. The resulting flow cost is equal to 28 ($10 \cdot 2 = 20$ for the



k	o_k	s_k	d_k	t_k^{MAX}	α_k	β_k
1	A	D	2	1	4	0
2	B	A	2	2	3	0

Fig. 2. Network and arc costs (left) and commodities specification (right).

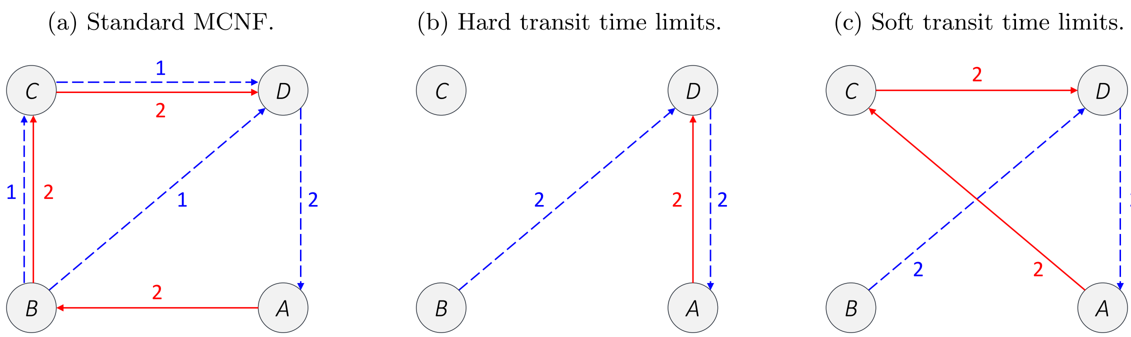


Fig. 3. Optimal flow solution for different MCNF models. The flows of commodity 1 and 2 are represented, respectively, by the continuous red line and the dashed blue line (with related values).

first commodity and $4 \cdot 2 = 8$ for the second), which is substantially higher than the cost of the (unconstrained) MCNF solution from Fig. 3(a).

We finally consider the STC-MCNF with penalties defined by the parameters in Fig. 2, and show the corresponding optimal solution in Fig. 3(c). The first commodity takes the path ACD that consumes two units of time. This path has a unit cost of 5, is subject to a unit penalty of 4 (the delay is equal to 1), and is consequently preferable to its alternatives, i.e.:

- the longer path $ABCD$ used in the MCNF solution (see Fig. 3(a)), with a total arc cost of only 3 but a larger penalty of 8 since the delay for this path is equal to 2;
- the direct connection AD used in the HTC-MCNF solution (see Fig. 3(b)), with a total arc cost of 10 and no penalty.

The cost of flowing the first commodity is hence 18 (=9·2). Regarding the second commodity, due to the penalty function, the path BDA with no delay becomes cheaper than path $BCDA$ with a delay of 1. The objective function is 26 (the cost is 18 for the first commodity and $4 \cdot 2 = 8$ for the second), which is about 7% lower than the solution from the HTC-MCNF model. Thus, there is a benefit in solving the STC-MCNF because: (i) it accounts for transit time and delays, a feature that is missing in the standard MCNF, and (ii) it provides a much higher flexibility than the HTC-MCNF that may lead to overall cheaper solutions.

3.3. Properties of STC-MCNF and relationships with other models

Next we show some properties and relationships that hold among the different multi-commodity flow models. In the first part of this section, we will refer to the same example from Section 3.2.

First of all, notice that the optimal solutions to both the MCNF and HTC-MCNF are always feasible to STC-MCNF, but might be suboptimal. Indeed, if we consider the MCNF solution in Fig. 3(a) and subsequently evaluate the penalty to each path in this solution, we obtain a total cost of 32 ($[3+8]\cdot 2 = 22$ for the first commodity and $4+3+3 = 10$ for the second), which is about 26% higher than the STC-MCNF solution of 26. This approach can be seen as a simple way to obtain a feasible solution to the STC-MCNF without accounting for transit time explicitly. We henceforth refer to this approach for the STC-MCNF as the *sequential heuristic* (SH).

The following proposition shows that the cost discrepancy between the optimal solutions from the STC-MCNF and both the HTC-MCNF and SH can be arbitrarily large.

Proposition 1 (*Cost Divergence*)**.**

(a) Consider the collection of instances indexed by $\xi \in \mathbb{R}_+$, obtained from the example in Section 3.2 by varying the arc cost $c_{AD} := \xi$. The difference between the optimal HTC-MCNF and STC-MCNF cost increases indefinitely as $\xi \to +\infty$.

(b) Consider the collection of instances indexed by $\xi \in \mathbb{R}_+$, obtained from the example in Section 3.2 by varying the penalty slope $\alpha_2 := \xi$. The difference between the SH cost and the optimal STC-MCNF cost increases indefinitely as $\xi \to +\infty$.

Proof. Part (a). In the HTC-MCNF optimal solution, commodity 1 flows through AD regardless of its cost $c_{AD} = \xi$ since only this path respects the hard transit time limit. Consequently, the HTC-MCNF optimal flow corresponds to the one displayed in Fig. 3(b) for any $\xi \in \mathbb{R}_+$ and has a cost of $8 + 2\xi$. Moreover, it is easy to verify that for any $\xi \geqslant 9$ the optimal STC-MCNF flow is the one reported in Fig. 3(c) with a cost of 26. Thus, for large enough ξ, the cost difference when using hard and soft transit time constraints is $8 + 2\xi - 26 \to +\infty$ as $\xi \to +\infty$.

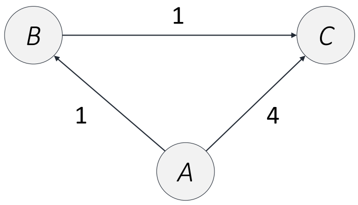
Part (b). The flow in the SH solution is not affected by the penalty functions, hence it matches with Fig. 3(a) for any $\xi \in \mathbb{R}_+$. Following the steps described in the example, the cost of this solution is $29 + \xi$. Moreover, the optimal STC-MCNF flow for $\xi \geqslant 1$ corresponds to the one shown in Fig. 3(c). Its objective value is 26 and is not affected by ξ. Thus, the difference between the former and the latter costs is $29 + \xi - 26 = 3 + \xi \to +\infty$ as $\xi \to +\infty$. □

Proposition 1 highlights the relevance of considering penalty functions and hence soft transit time constraints explicitly when routing cargo, especially time sensitive cargo. Part (a) of this proposition may represent the scenario in which the cost of a network link increases, for example due to increasing tolls to cross the Panama or Suez canal. Part (b) may instead represent a situation in which the time sensitivity of goods suddenly changes, as e.g. has been the case with medical equipment during the Covid-19 pandemic (Bloomberg, 2020).

Since penalty functions translate in the end to additional costs, one may ask whether solving the STC-MCNF is actually needed rather than instead increasing/modifying the arc costs in the network and solving a standard MCNF. Proposition 2 addresses this question and shows that solving the STC-MCNF and MCNF problems is not equivalent, i.e., it is not possible to solve the STC-MCNF as a standard MCNF in an augmented network with arbitrarily altered arc costs. This proposition provides indeed further theoretical backing for studying the STC-MCNF.

Proposition 2 (*Arc Cost Modification*)**.** *Given an STC-MCNF instance, there does not exist in general a modification of the arc costs $\tilde{c} \in \mathbb{R}^{|\mathcal{A}|}$ such that the sets of optimal solutions from the standard MCNF in the modified graph and the STC-MCNF in the original graph coincide.*

Proof. The proof is based on a simple counter-example of a network with three nodes and two commodities sharing origin and destination but subject to different piecewise linear penalties (see Fig. 4). The figure reports the arc unit costs c_{ij} while the capacities u_{ij} and transit times t_{ij} (omitted in the figure) are the same for all arcs and are equal to 2 and 1, respectively.



k	o_k	s_k	d_k	t_k^{MAX}	α_k	β_k
1	A	C	1	1	4	0
2	A	C	1	2	3	0

Fig. 4. Network and arc costs (left) and commodities specification (right).

It is easy to see that there is a unique optimal STC-MCNF solution where commodities 1 and 2 follow paths AC and ABC, respectively. In the standard MCNF, instead, both commodities will use the same cheaper route regardless of the cost modification (capacity is non-binding, hence the problem can be decomposed by commodity). More precisely, for cost modifications such that $\widetilde{c}_{AC} < \widetilde{c}_{AB} + \widetilde{c}_{BC}$, both commodities will use path AC. Instead, when $\widetilde{c}_{AC} > \widetilde{c}_{AB} + \widetilde{c}_{BC}$, both commodities will use path ABC. Finally, in the special case in which $\widetilde{c}_{AC} = \widetilde{c}_{AB} + \widetilde{c}_{BC}$, the MCNF optimal solution set includes four different solutions (i.e., all combinations of the two commodities flowing along the two paths), so again this set differs from the STC-MCNF singleton. □

We conclude this section by discussing the complexity of STC-MCNF.

Proposition 3 (*Complexity*). *The STC-MCNF is in general $\mathcal{NP}$-hard in the strong sense. If arc transit times are integer values, i.e. $t_{ij} \in \mathbb{N}_+$, then the STC-MCNF becomes weakly $\mathcal{NP}$-hard.*

Proof. The $\mathcal{NP}$-hardness of STC-MCNF is shown by reduction from the RCSP, that is known to be $\mathcal{NP}$-hard (Garey and Johnson, 1990). Given an instance of the RCSP with source s, destination d and time resource limit t^{MAX}, we can transform it to an instance of the STC-MCNF having only one commodity $k = 1$ with the same source $s_k = s$ and destination $d_k = d$. Moreover, we use a piecewise constant penalty function as the one in Fig. 1(a) with a zero penalty for all $t \leqslant t^{\text{MAX}}$ and a very large (infinite) penalty for all $t > t^{\text{MAX}}$. Clearly an optimal solution to the STC-MCNF instance, is also an optimal solution to the RCSP.

The STC-MCNF becomes weakly $\mathcal{NP}$-hard when we use integer arc transit times, in other words, it becomes solvable in pseudo-polynomial time. This follows immediately from STC-MCNF admitting a linear programming formulation with model size that depends on one of the coefficients in the problem, namely an upper bound T^{UP} on commodity transit times (see Online Supplement B). Alternatively, the pseudo-polynomial complexity derives from the RCSP becoming weakly $\mathcal{NP}$-hard when time is discrete (Hassin, 1992). This is evident in Section 4.2 as the number of labels for a node processed during the dynamic programming algorithm is bounded by the number of time steps. □

4. Path-flow formulation and column generation

In this section we present a path-flow formulation and a CG method to solve the STC-MCNF. In Section 4.1, we introduce essential notations and formulate the master problem. We then discuss the pricing problem for the HTC-MCNF and the STC-MCNF in Sections 4.2 and 4.3, respectively.

4.1. Master problem

In the path-flow formulation, we want to determine how much of commodity k to send along paths in the network that start in o_k and end in s_k. Formally, for each commodity $k \in \mathcal{K}$, we denote by $\mathcal{P}^k$ the set of feasible and simple paths from o_k to s_k. Choosing non-simple paths is never optimal because these paths have higher cost and capacity utilization compared to the same paths where the loops are removed, hence non-simple paths can be discarded. We denote with $\mathcal{P} = \bigcup\{\mathcal{P}^k, k \in \mathcal{K}\}$ the complete set of simple paths for all commodities, and with $\mathcal{P}^k_{(i,j)} := \{p \in \mathcal{P}^k : (i,j) \in p\}$ the subset of commodity-k paths which contain the arc $(i,j) \in \mathcal{A}$. The cost c_p of a path $p \in \mathcal{P}$ is given by summing up the cost of the arcs that are part of it, i.e. $c_p = \sum\{c_{ij} : (i,j) \in p\}$. For a commodity $k \in \mathcal{K}$ and a path $p \in \mathcal{P}_k$, the modified cost $c'_p := c_p + f_k(t_p)$ accounts for the penalty function, where $t_p = \sum\{t_{ij} : (i,j) \in p\}$ is the total transit time needed to traverse the path. The decision variables $x^k_p \geqslant 0$ represents the amount of commodity k that is sent along path p. The master problem is formulated as the following linear program:

$$\min \sum_{k\in\mathcal{K}} \sum_{p\in\mathcal{P}^k} c'_p x^k_p \tag{2a}$$

$$\text{s.t.:} \sum_{p\in\mathcal{P}^k} x^k_p = d_k \qquad \forall k \in \mathcal{K}, \tag{2b}$$

$$\sum_{k\in\mathcal{K}} \sum_{p\in\mathcal{P}^k_{(i,j)}} x^k_p \leqslant u_{ij} \qquad \forall (i,j) \in \mathcal{A}, \tag{2c}$$

$$x^k_p \geqslant 0 \qquad \forall k \in \mathcal{K},\ p \in \mathcal{P}^k. \tag{2d}$$

The objective function (2a) is the penalized cost of the flow, constraints (2b) ensure that the commodity demand is satisfied, and constraints (2c) enforce the network capacity limits. Flow conservation is automatically ensured with path-flow variables. The master problem (2) resembles a set covering model; the path-flow formulation of MCNF coincides in fact with master problem (2) when replacing the penalized cost c'_p with c_p in the objective function (2a) for all paths $p \in \mathcal{P}$.

It is well known that directly formulating and solving the master problem (2) is intractable due to the exponential number of routes/variables, hence CG should be used instead. We assume the reader is familiar with CG and therefore only give a brief introduction to CG for MCNF in Online Supplement A. The restricted master problem in CG is a reduced version of the master problem (2) only containing a (small) subset of paths $\widetilde{\mathcal{P}} \subseteq \mathcal{P}$, and the pricing problem (or subproblem) implicitly seeks for a negative-reduced-cost path to add to the master problem. Although the master problem is easy to formulate as a linear program, the $\mathcal{NP}$-hard complexity of STC-MCNF is now hidden in the path generation subproblem.

The subproblem for a commodity k requires determining the path from o_k to s_k that minimizes its reduced cost, including the sum of arc costs and delay penalty that depends on the total transit time duration. This problem cannot be reformulated as a standard shortest path since the penalty $f_k(t_p)$ is a function of the total transit time t_p, hence $f_k(t_p)$ cannot be repartitioned among the arcs in the form of cost. In other words, our objective is *non-additive* (Reinhardt and Pisinger, 2011). Instead, this problem is related to the RCSP, in which arcs $(i,j) \in \mathcal{A}$ are associated with the consumption of two resources: an *unconstrained* resource that has to be minimized and a *constrained* resource for which a certain threshold must not be exceeded. This problem is well know in the operations research literature (see, e.g., Irnich and Desaulniers, 2005) and arises as a CG subproblem for the HTC-MCNF (Holmberg and Yuan, 2003; Karsten et al., 2015; Koza et al., 2020). We thus introduce first the case of HTC-MCNF and then discuss how to adapt it to the STC-MCNF in Section 4.3.

4.2. Solving the subproblem in the case of HTC-MCNF

In the HTC-MCNF subproblem we want to determine, for each $k \in \mathcal{K}$, the path from o_k to s_k that minimizes the reduced cost and such that the transit time duration is below or equal to the threshold t_k^{MAX}. Given path $p \in \mathcal{P}_k$ and dual values η_k and γ_{ij} associated with constraints (2b) and (2c), respectively, the reduced cost $\bar{c}_p$ of the path is $\bar{c}_p = \sum_{(i,j)\in p} \left(c_{ij} - \gamma_{ij}\right) - \eta_k$. Thus, for each arc $(i,j) \in \mathcal{A}$, the unconstrained resource is the cost $\bar{c}_{ij} = c_{ij} - \gamma_{ij}$ and the constrained resource is the transit time t_{ij}. We solve this problem using a label-correcting algorithm based on dynamic programming (Irnich and Desaulniers, 2005) that we outline below focusing first on a single commodity k. We remark that this is a common approach used in other works on HTC-MCNT (Holmberg and Yuan, 2003; Karsten et al., 2015; Koza et al., 2020) but different methods also exist for solving the RCSP as mentioned in the literature review in Section 2.1.

For a node $i \in \mathcal{N}$, we define a *label* for i as a pair $(C_i, T_i) \in \mathbb{R}^2_+$, where C_i and T_i represent, respectively, the reduced cost and the cumulative transit time of a path connecting the source o_k with the node i. A node can be associated with many labels $\mathcal{L}_i = \{(C_i^h, T_i^h),\ h = 1, \dots, L_i\}$, where L_i denotes the number of labels, representing multiple ways of reaching node i from o_k utilizing different amount of resources. Initially, the origin node o_k is provided with a single label $(0,0)$. Moreover, a queue of unprocessed nodes $\mathcal{N}^{\text{U}}$ is introduced and initialized to be the entire node set $\mathcal{N}$, with the first element being o_k. At a high level, the algorithm proceeds by (i) determining labels for a node that are Pareto-optimal in terms of reduced cost and transit time, and (ii) extending these labels to all the outgoing arcs of the current node. More specifically, at each iteration the algorithm picks the first node in the queue $i \in \mathcal{N}^{\text{U}}$ and removes it from the queue, i.e., $\mathcal{N}^U \leftarrow \mathcal{N}^U \setminus \{i\}$. The labels $\mathcal{L}_i$ of node i propagate to all its successor nodes j by extension through the outgoing arcs. Formally, for a successor node j, the label extension function produces $\mathcal{L}_j \leftarrow \mathcal{L}_j \cup \{(C_i^h + \bar{c}_{ij}, T_i^h + t_{ij}),\ h = 1, \dots, L_i\}$. After this process is completed, the labels in the updated set $\mathcal{L}_j$ which are not Pareto-optimal are deleted from $\mathcal{L}_j$ because obviously they correspond to suboptimal (sub-)paths terminating at node j. Thus, keeping and extending these labels would just be a waste of computational resources. We say that these labels can be fathomed according to the following definition.

⋄ **Fathoming** $\mathfrak{F}1$ [Dominance]. *Label* (C_i^h, T_i^h) *can be* $\mathfrak{F}1$*-fathomed if there is another label* (C_i^l, T_i^l) *for the same node such that* $C_i^l \leqslant C_i^h$ *and* $T_i^l \leqslant T_i^h$.

Notice that fathoming by dominance requires that consumption of both resources is non-decreasing in each label extension, which is true in our case since $\bar{c}_{ij} \geqslant 0$ and $t_{ij} \geqslant 0$ for each $(i,j) \in \mathcal{A}$. Furthermore, labels of a node for which the cumulative transit time exceeds the limit t_k^{MAX} can also be discarded because they represent infeasible paths for HTC-MCNF. We define this rule as follows:

⋄ **Fathoming** $\mathfrak{F}2$ [HTC]. *Label* (C_i^h, T_i^h) *can be* $\mathfrak{F}2$*-fathomed if* $T_i^h > t_k^{\text{MAX}}$.

When a new potentially useful label is added to $\mathcal{L}_j$, node j is (re-)inserted in the unprocessed node queue $\mathcal{N}^{\text{U}} \leftarrow \mathcal{N}^{\text{U}} \cup \{j\}$, if not already there. The algorithm terminates when all nodes are processed, i.e., when $\mathcal{N}^{\text{U}} = \emptyset$. The output consists of a list of labels $\mathcal{L}_i = \{(C_i^h, T_i^h),\ h = 1, \dots, L_i\}$ for each node $i \in \mathcal{N}$, and the label for the sink node s_k with minimum reduced cost is chosen. Finally, the path associated with this label can be reconstructed by backtracking, i.e. starting from node i and tracking the labels backward to determine the previous nodes in the path until the origin o_k is reached. When transit times are integer, the runtime of this label-correcting algorithm is pseudo-polynomial as the number of labels for a node is bounded by t_k^{MAX}.

In addition to $\mathfrak{F}1$ and $\mathfrak{F}2$, previous research has identified improved fathoming criteria that help reducing the number of labels further. For each node i, one can compute $\bar{c}_{i\to s_k}^{\text{MIN}}$ and $t_{i\to s_k}^{\text{MIN}}$ representing the length of the (standard) shortest path from i to s_k that only minimizes, respectively, reduced cost and transit time alone. These quantities can be computed in polynomial time in a preprocessing phase. With them, we can define:

⋄ **Fathoming** $\mathfrak{F}3$ [Time]. *Label* (C_i^h, T_i^h) *can be* $\mathfrak{F}3$*-fathomed if* $T_i^h + t_{i\to s_k}^{\text{MIN}} > t_k^{\text{MAX}}$.

⋄ **Fathoming** $\mathfrak{F}4$ [Cost]. *Label* (C_i^h, T_i^h) *can be* $\mathfrak{F}4$*-fathomed if* $C_i^h + \bar{c}_{i\to s_k}^{\text{MIN}} > \eta_k$.

$\mathfrak{F}3$ states that, despite the current transit time T_i^h is feasible, extending this label to s_k would necessarily exceed t_k^{MAX} as it would take at least $t_{i\to s_k}^{\text{MIN}}$ units of time more to reach s_k. Notice that $\mathfrak{F}2 \Rightarrow \mathfrak{F}3$, that is, $\mathfrak{F}3$-fathoming is stronger than $\mathfrak{F}2$-fathoming in the sense that it is able to potentially discard many more labels. Similarly, $\mathfrak{F}4$ states that extending C_i^h to s_k would result in a positive reduced cost path, which is not useful in our column generation context and can be discarded.

As discussed, the label-correcting algorithm not only produces labels for the sink node s_k but also, as by-product, labels for all nodes $i \in \mathcal{N}$. Therefore, instead of solving the RCSP for each commodity $k \in \mathcal{K}$, one could group commodities sharing the same source node and solve the RCSP only once per group. This strategy, called *one-to-all* RCSP, is particularly attractive when the number of commodities is much larger than the number of nodes from which at least one commodity departs, which we denote by origin nodes $\mathcal{N}^O \subseteq \mathcal{N}$. Let $\mathcal{K}_j := \{k \in \mathcal{K} : o_k = j\} \subseteq \mathcal{K}$ be the subset of commodities sharing j as origin. The fathoming rules $\mathfrak{F}2$, $\mathfrak{F}3$, and $\mathfrak{F}4$ are commodity-specific and must be relaxed in order to be used in the one-to-all setting. Thus, we define:

⋄ **Fathoming** $\mathfrak{F}2_R$ [HTC-R]. *Label* (C_i^h, T_i^h) *can be* $\mathfrak{F}2_R$*-fathomed if* $T_i^h > \max\{t_k^{\text{MAX}},\ k \in \mathcal{K}_j\}$.

⋄ **Fathoming** $\mathfrak{F}3_R$ [Time-R]. *Label* (C_i^h, T_i^h) *can be* $\mathfrak{F}3_R$*-fathomed if* $T_i^h + \min\{t_{i\to s_k}^{\text{MIN}},\ k \in \mathcal{K}_j\} > \max\{t_k^{\text{MAX}},\ k \in \mathcal{K}_j\}$.

⋄ **Fathoming** $\mathfrak{F}4_R$ [Cost-R]. *Label* (C_i^h, T_i^h) *can be* $\mathfrak{F}4_R$*-fathomed if* $C_i^h + \min\{\bar{c}_{i\to s_k}^{\text{MIN}},\ k \in \mathcal{K}_j\} > \max\{\eta_k,\ k \in \mathcal{K}_j\}$.

Clearly, these fathoming rules are weaker than their respective single-commodity version. Specifically, $\mathfrak{F}2_R \Rightarrow \mathfrak{F}2$, $\mathfrak{F}3_R \Rightarrow \mathfrak{F}3$, and $\mathfrak{F}4_R \Rightarrow \mathfrak{F}4$. Therefore, there is a trade-off here between the quality of the fathoming criteria and the number of subproblems to solve. In the HTC-MCNF literature, authors dealing with column generation have identified and/or chosen different sets of fathoming rules in the RCSP subproblem, as we report in Table 1.

Table 1
Use of RCSP fathoming rules in previous HTC-MCNF research.

Paper	Type of subproblem	Fathoming rules
Holmberg and Yuan (2003)	Single-commodity	$\mathfrak{F}1 + \mathfrak{F}3 + \mathfrak{F}4$
Karsten et al. (2015)	One-to-all	$\mathfrak{F}1 + \mathfrak{F}2_R$
Koza et al. (2020)	One-to-all	$\mathfrak{F}1 + \mathfrak{F}3_R + \mathfrak{F}4_R$

4.3. STC-MCNF adaptation and goal-based fathoming rules

Although the STC-MCNF has no hard transit time limits, the RCSP algorithm detailed in Section 4.2 can be adapted to solve the subproblem arising in the STC-MCNF CG algorithm as follows.

First, in contrast to the HTC-MCNF, when the algorithm terminates, we select the label for a node i (e.g., the sink node $i = s_k$) that minimizes the sum of reduced cost and penalty function, i.e. $(C_i^*, T_i^*) := \operatorname{argmin}\left\{C_i^h + f_k(T_i^h) : h = 1, \ldots, L_i\right\}$.

Second, we only add the chosen path(s) to the master problem if a "penalized" reduced cost is negative, which is defined as

$$\bar{c}_p = \sum_{(i,j)\in p} (c_{ij} - \gamma_{ij}) - \eta_k + f_k\left(\sum_{(i,j)\in p} t_{ij}\right). \tag{3}$$

Third, the fathoming criteria have to be revisited in order to be applied. Recall that fathoming labels is critical to keep the total number of active labels small and hence limit computational memory and time. Fathoming by dominance ($\mathfrak{F}1$) as well as the rules involving reduced cost alone ($\mathfrak{F}4$ and $\mathfrak{F}4_R$) remain applicable in the context of STC-MCNF. Instead, fathoming rules involving transit time ($\mathfrak{F}2$, $\mathfrak{F}2_R$, $\mathfrak{F}3$, and $\mathfrak{F}3_R$) require replacing t_k^{MAX} with an upper bound T_k^{UP} large enough to ensure that no optimal path to STC-MCNF is discarded (see also Online Supplement B where we discuss how T_k^{UP} can be defined). This modification makes transit time-related rules much weaker.

To compensate this shortcoming, we adapt to our subproblem (the single-commodity case) the so called goal-based fathoming criteria (Reinhardt and Pisinger, 2011), which have not been used in MCNF-related research. In particular, unless $\mathcal{G}$ is acyclic, each node $i \in \mathcal{N}$ is usually visited several times before the label-correcting algorithm terminates. After the sink node s_k is visited for the first time, it is provided with a set of labels corresponding to complete paths in the graph for the commodity k. The idea is to exploit labeling information that becomes available for s_k to define a fathoming rule valid for all other nodes and that is based on a comparison with the labels of s_k.

⋄ **Fathoming** $\mathfrak{F}5$ [Goal]. *Label* (C_i^h, T_i^h) *can be* $\mathfrak{F}5$*-fathomed if there exists a sink-node label* $l = (C_{s_k}^l, T_{s_k}^l) \in \mathcal{L}_{s_k}$ *such that* $C_i^h + \bar{c}_{i\to s_k}^{\text{MIN}} + f(T_i^h + t_{i\to s_k}^{\text{MIN}}) \geqslant C_{s_k}^l + f(T_{s_k}^l)$.

The condition in $\mathfrak{F}5$ means that extending the label (C_i^h, T_i^h) to s_k would result in a complete path with penalized reduced cost which is at least as high as the penalized reduced cost of a path that has already been found. Therefore, this label can be discarded. In addition to reduced cost and transit time being positive, the $\mathfrak{F}5$-fathoming requires the penalty function to be non-decreasing (see our related discussion at the end of Section 3.1). Clearly, this rule works specifically for the STC-MCNF as it makes explicit use of the penalty function. However, the following relaxed formulation of it can be used in the generic RCSP algorithm, e.g., for the HTC-MCNF subproblem.

⋄ **Fathoming** $\mathfrak{F}6$ [Goal-General]. *Label* (C_i^h, T_i^h) *can be* $\mathfrak{F}6$*-fathomed if there exists a sink-node label* $l = (C_{s_k}^l, T_{s_k}^l) \in \mathcal{L}_{s_k}$ *such that* $(C_i^h + \bar{c}_{i\to s_k}^{\mathrm{MIN}}, T_i^h + t_{i\to s_k}^{\mathrm{MIN}})$ *can be* $\mathfrak{F}1$*-fathomed due to* l.

It holds that $\mathfrak{F}6 \Rightarrow \mathfrak{F}5$, i.e., $\mathfrak{F}5$ is tighter than $\mathfrak{F}6$.

Even though it is possible in principle to define a one-to-all version of $\mathfrak{F}5$, such relaxation would be extremely unlikely to be triggered as it would require all destination nodes s_k to have been visited at least once (which is instead not required for the other one-to-all relaxations $\mathfrak{F}3_R$ and $\mathfrak{F}4_R$), in addition to the condition described in $\mathfrak{F}5$ to hold simultaneously for each of these nodes. For this reason, we do not consider a one-to-all relaxation of $\mathfrak{F}5$. To summarize, the strongest set of fathoming rules in the STC-MCNF subproblem can be defined as follows:

$$\begin{cases} \mathfrak{F}1 + \mathfrak{F}3 + \mathfrak{F}4 + \mathfrak{F}5 \text{ (single-commodity)}, & \text{if } K \leqslant \mu \, \mathcal{N}^O, \\ \mathfrak{F}1 + \mathfrak{F}3_R + \mathfrak{F}4_R \text{ (one-to-all)}, & \text{otherwise}, \end{cases}$$

where the parameter μ establishes when single-commodity rather than one-to-all subproblems should be solved based on the number of commodities and origin nodes.

Notice that, analogously to the standard RCSP from Section 4.2, this variant of the problem is also strongly $\mathcal{NP}$-hard and becomes weakly $\mathcal{NP}$-hard under the integer transit time assumption, which is consistent with Proposition 3.

5. Study design

In this section we introduce our experimental setup based on cargo routing in liner shipping. This includes defining our instances (Section 5.1), constructing the networks (Section 5.2), choosing the penalty functions (Section 5.3), and specifying the computational settings (Section 5.4).

5.1. Liner shipping instances

We test our models on the LINER-LIB benchmark instances (Brouer et al., 2013a). Since the LINER-LIB describes unscheduled networks that serve as basis for solving the liner shipping network design problem, we cannot use them directly but rather we use some of their network design solutions. Specifically, we consider the instances {WAF, MED, PAC, WS, EUA} representing different geographical regions, and for each of them, we test 12 different scheduled networks resulting as output from the matheuristic of Koza et al. (2020). These networks indeed represent the best-known LINER-LIB network design solutions. For clarity, we henceforth call the five LINER-LIB networks *classes*, or instance classes, and each specific network design solution an *instance*. Therefore, in our experiments we consider a total of $5 \cdot 12 = 60$ instances.

In Table 2, we present the five LINER-LIB classes with network size and number of commodities averaged over their 12 respective instances. The five classes are sorted by increasing size and range from small-sized instances {WAF} to medium {MED, PAC} and large instances {WS, EUA}.

A scheduled liner shipping network represents essentially a collection of *services*, where a service is a sailing route operated at a weekly frequency. Next we explain how, starting from these services, we have obtained the networks $\mathcal{G} = (\mathcal{N}, \mathcal{A}, c, u, t)$ used in our numerical experiments.

Table 2
Instance classes with average network size and number of commodities.

Class	Description	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{N}^O\rvert$	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{K}\rvert$
WAF	West Africa	69.5	16	270.9	30.1
MED	Mediterranean	101.6	26.1	498.1	247.3
PAC	Pacific region	189.7	34.2	1567.7	581.7
WS	World small	356.1	44.8	4312.5	1615.9
EUA	Europe Asia	353.7	67.0	4313.7	2428.8

5.2. Network construction

Set of nodes $\mathcal{N}$: For each port X that is served by at least one service, we introduce a set of nodes $X_1, \ldots, X_n$, that represent all n port calls of port X. Each port call and hence each node is associated with a weekly arrival time. Moreover, for each port X from which at least a commodity $k \in \mathcal{K}$ departs, we create a *source node* X_o for that port, and for each port X that is destination for at least one commodity $k \in \mathcal{K}$, we create a *sink node* X_s.

Set of arcs $\mathcal{A}$: Each service already defines a set of *sailing arcs* representing the sailing of specific liner shipping services between two scheduled port calls. The capacity of each arc corresponds to the capacity of the vessel type that is deployed on a service. There could be two or more services that sail between the same ports at the same time, and they would be represented by different sailing arcs. The travel cost of a twenty-foot equivalent (TEU) container on a sailing arc corresponds to the marginal increase in

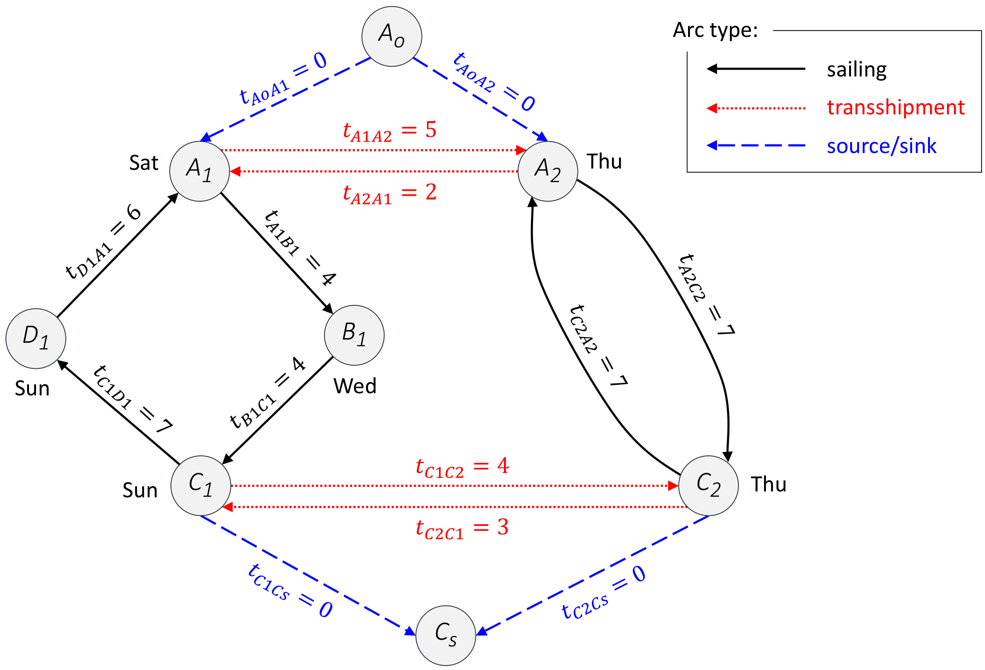


Fig. 5. Graph $\mathcal{G}$ of associated problem.

fuel cost caused by the additional load of a container, assuming an average weight of 11 tons. The cost varies between arcs and depends on the vessel type and the sailing speed associated with that arc. The formula to calculate this cost for different vessel speeds and types, and its validation can be found in Koza (2019). Note that the penalties/rewards for late/early arrivals are added to the transportation costs.

Moreover, there exist *transshipment arcs* that connect different services visiting the same port. For instance, if port X is called by two different services represented by port call nodes X_1 and X_2, then containers can be transshipped between the two port calls via the arcs (X_1, X_2) and (X_2, X_1). In addition to the sailing and transshipment arcs, we introduce a set of *source arcs* connecting each source node X_o with all nodes representing port calls $X_1, \ldots, X_n$ of port X. Analogously, for each sink node X_s, we introduce *sink arcs* connecting $X_1, \ldots, X_n$ to X_s. All source and sink arcs have a transit time duration of zero, infinite capacity, and a cost equal to the container unit move cost of the corresponding port.

To illustrate this construction, consider the example shown in Fig. 5. This network represents 2 services operating across 4 ports: A, B, C, and D. Service 1 calls ports $ABCDA$ with a total duration of 21 days, i.e. three vessels are deployed on that service to achieve a weekly frequency. Service 2 calls ports ACA with a total duration of 14 days, i.e. two vessels are deployed on that service. The nodes in the associated graph represent port calls and are labeled with the weekday on which the port is called (e.g., A_1 and A_2 denote two calls of port A, respectively on Saturday by service 1 and on Thursday by service 2). Arcs between two port call nodes are labeled with the transit time duration, e.g., sailing from A_1 to B_1 takes $t_{A_1B_1} = 4$ days and transshipment from A_2 to A_1 takes $t_{A_2A_1} = 2$ days. There is one commodity that needs to be shipped from port A to C. Thus, we introduce in the graph one origin node A_o and one destination node C_s for that commodity, and the corresponding source arcs (A_o, A_1), (A_o, A_2), and sink arcs (C_1, C_s), (C_2, C_s), each provided with transit time equal to zero.

Setting the transit time durations of the source arcs (A_o, A_1) and (A_o, A_2) equal to zero can be interpreted as the customer being indifferent about the weekday of cargo pick-up. In other words, the customer ships the same amount of containers each week, and can deliver the weekly batch on any day of the week. The customer does, however, care about the total transportation time. Assume for instance that: (i) there is a strictly increasing delay penalty function, i.e. shorter transit time durations are always preferred, and (ii) arc traversal costs and capacities are respectively set to zero and infinity. Then, the optimal commodity flow follows the shortest path defined in travel time, which is $A_oA_2C_2C_s$. This path uses service 2 and has a total duration of 7 days.

A variant of the problem that we could also model using the STC-MCNF is when customers have the cargo ready for pick-up on a particular weekday. We do not consider this variant in our experiments but show in Online Supplement C the network corresponding to Fig. 5 in this case.

5.3. Definition of penalty functions

Our reference penalty function is based on a recent liner shipping publication (Reinhardt et al., 2020) which presents a functional form and parameters that have been validated by a major liner shipping operator. To obtain meaningful business insights, we therefore adopt shape and parameters in our experiments. The proposed penalty function is piecewise linear with one slope for modeling delays and one for modeling early arrivals as shown in the example in Fig. 1(b). Defining such penalty function requires specifying, for each commodity $k \in \mathcal{K}$, the slope of extra costs incurred in case of delays (α_k), the slope of early arrival bonuses (β_k), and the reference arrival time at which the penalty equals zero (t_k^{MAX}) that also serves as hard transit time limit for the HTC-MCNF. We compute these parameters in four steps:

1. We solve the standard MCNF – without hard or soft transit time constraints – using CG to obtain a baseline. We recall that CG for the standard MCNF involves solving a master problem that is essentially the same as (2) for the HTC/STC-MCNF but with a different subproblem, i.e., a standard shortest path problem instead of an RCSP variant (see Online Supplement A for an introduction to the standard MCNF and related CG solution method). This solution contains, for each commodity k, the set of paths $\mathcal{P}_1^{k,*} \subseteq \mathcal{P}^k$ that the commodity takes, with their respective cost c_p and transit time t_p. Clearly, it can happen that $|\mathcal{P}_1^{k,*}| \geqslant 1$.
2. We solve a standard MCNF as we do in step 1, but in which arc costs c_{ij} are replaced by their transit time durations t_{ij}. This gives a second set of paths $\mathcal{P}_2^{k,*} \subseteq \mathcal{P}^k$ used by commodities, that we expect on average to have shorter transit time durations than the former paths $\mathcal{P}_1^{k,*}$.
3. Given $t_k^1 := \max\{t_p, p \in \mathcal{P}_1^{k,*}\}$ and $t_k^2 := \max\{t_p, p \in \mathcal{P}_2^{k,*}\}$, we set $t_k^{\text{MAX}} = \lceil \gamma_k t_k^1 + (1-\gamma_k) t_k^2 \rceil \gamma_k t_k^1 + (1-\gamma_k) t_k^2$ if $t_k^1 > t_k^2$ and $t_k^{\text{MAX}} = t_k^1$ otherwise, with $\gamma_k \in [0,1]$. Thus, t_k^{MAX} is defined as a convex combination of t_k^1 and t_k^2 governed by the parameter γ_k. With this choice, we know that at least some commodities would benefit from flowing faster than in the MCNF and at the same time the HTC-MCNF is feasible.
4. We set $\alpha_k = 25$ USD and $\beta_k = 2.5$ USD according to Reinhardt et al. (2020)[1], and $\gamma_k = 0.8$.

In Sections 6.2–6.4, we also analyze penalties obtained from the reference described above by varying α_k (we call them penalties **P1**), β_k (penalties **P2**) and γ_k (penalties **P3**) from their respective baseline values, where the effect of increasing and decreasing γ_k, respectively, is to increase and decrease t_k^{MAX}. We further consider piecewise constant (i.e. staircase) penalty functions in Section 6.5.

5.4. Computational setup

Models and algorithms were implemented in C++ using Gurobi 9.1 as linear programming solver for the CG master problem. The experiments were run on a standard laptop equipped with an i7-8650U processor and consumed at most 1 GB RAM.

We initialize CG using for each commodity $k \in \mathcal{K}$ a dummy path that directly connects o_k with s_k with large cost and transit time values (big-M). While running CG, the master problem at each iteration is warm started using the solution from the previous iteration. Our RCSP based pricing allows to pick multiple violated columns per commodity to enter the master problem in a single iteration. However, we found that the running time when adding more columns was almost identical to the case of adding a single column per commodity. Therefore, the results we present are based on generating and adding one column per commodity per iteration. Since CG in our application is quick and always converges in a few iterations (less than 20–25), stabilization techniques that could potentially accelerate the process appear not needed here, hence they were not investigated.

Since a flow denotes a number of containers, technically speaking our application would be an integer MCNF that would require embedding CG in a branch-and-price framework to be solved optimally. However, typical liner shipping problems involve thousands of containers on a single vessel and research has shown that retrieving integer flows by rounding down fractional components leads to negligible approximation errors (Brouer et al., 2011). We checked whether fractionality plays a role in our numerical results by solving the (continuous) STC-MCNF and investigating the amount of fractional flows in the optimal solution. On average across our 60 instances we found that: (i) the aggregate amount of fractional flows constitute less than 0.02% of the total commodity demand, and (ii) the cost of the integer flows obtained by rounding down fractional quantities is 0.008% away from the cost of the continuous solution. This means that the continuous and rounded-down-integer flows are almost identical and explicitly dealing with integrality constraints would not bring significant benefits. In our application, the weekly quantities are estimates that may vary slightly between weeks. Under the light of estimation errors, dealing with integrality becomes even more negligible.

We compared the STC-MCNF optimal solution against two benchmarks:

1. *The HTC-MCNF optimal solution*. To compute its objective value, we determine the optimal HTC-MCNF commodity flow using the hard transit times t_k^{MAX} from Section 5.3, and apply the penalty functions to capture discounts for early arrivals. Note that this latter step is necessary to evaluate the optimal STC-MCNF and HTC-MCNF solutions under the same objective.
2. *The SH solution*. To compute the SH objective value, we solve the standard MCNF, identify the paths chosen by each commodity and their respective transit times, and apply the corresponding penalties, which gives an objective function value for the STC-MCNF.

[1] Reinhardt et al. (2020) use $\alpha_k = 100/24$ USD per hour per FFE (forty-foot equivalent) container unit and $\beta_k = 10/24$ USD per hour per FFE. In our networks, flow-dependent arc cost is defined in USD per TEU and one unit of time (i.e. a transit time duration unit) is equal to 12 h (see Koza, 2019; Koza et al., 2020). Converting the rates from Reinhardt et al. (2020) to our setting gives the reported values for α_k and β_k.

Finally, we remark that our CG method to solve the path-flow formulation of the (continuous) STC-MCNF, as well as the standard MCNF and the HTC-MCNF, is exact and all instances are solved to optimality. In the following, our use of "gap/optimality gap" refers to objective function differences calculated with respect to the optimal objective function value of the STC-MCNF. Indeed, both the SH and the HTC-MCNF provide a feasible solution to the STC-MCNF, but these solutions may be suboptimal to the STC-MCNF; to quantify how much is one of our goals.

6. Results

In this section we discuss our results and related managerial insights. First, we compare the methods using the reference penalty function (Section 6.1). Then, we investigate the effect of varying the penalty coefficients on solution cost (Section 6.2) and commodity transit time (Section 6.3). Finally, we analyze the trade-off arising between the cost of the flow and average commodity transit time (Section 6.4), and the robustness of our findings towards the penalty functional choice (Section 6.5).

6.1. Comparison of solution methods by cost, gap, and runtime

In Table 3, we consider the reference penalty function and decompose the objective function value (i.e., total cost of the solution, label "obj.") into the cost associated with flows (labeled "flow") and with penalties (labeled "pen."). Assuming weekly commodity demands, the results are expressed in mln.USD per week of operation and are averaged over the 12 instances belonging to each class. We also report the CG running times in seconds (column "time") based on the single-commodity pricing strategy combined with the tightest set of fathoming rules (i.e., $\mathfrak{F}1$, $\mathfrak{F}3$, $\mathfrak{F}4$, and $\mathfrak{F}5$).

Table 3
Cost, gap, and running time of different formulations under the reference penalty.

Class	STC-MCNF				HTC-MCNF					SH				
	flow	pen.	obj.	time	flow	pen.	obj.	gap	time	flow	pen.	obj.	gap	time
WAF	3.79	−0.12	3.67	0.05	3.82	−0.13	3.69	0.49%	0.05	3.78	−0.08	3.71	0.85%	0.04
MED	2.48	0.06	2.53	0.08	2.56	−0.01	2.55	0.65%	0.08	2.42	0.25	2.67	5.25%	0.06
PAC	23.47	−0.92	22.54	0.61	23.64	−0.91	22.73	0.82%	0.55	22.89	0.85	23.74	5.30%	0.49
WS	83.26	0.35	83.61	4.30	87.14	−1.26	85.88	2.74%	4.19	81.64	6.18	87.82	5.08%	5.15
EUA	48.92	0.39	49.31	4.28	51.95	−0.96	51.00	3.44%	4.22	47.62	4.03	51.66	4.82%	5.92

In the case of HTC-MCNF, the flow component of the cost is higher than with STC-MCNF since commodities in the former model are routed via shorter and potentially more expensive paths. The penalty component in HTC-MCNF is instead always negative as paths linked to a strictly positive penalty are infeasible to this model. The resulting total cost is about 0.5% to 3.5% higher in HTC-MCNF, where this gap increases with the network size: the larger and more complex the network, the more routing options commodities can use in STC-MCNF to exploit penalties and discounts.

The situation is opposite with SH. The flow component of the cost with SH is smaller than that of STC-MCNF, which is expected since SH (based on the standard MCNF) optimizes indeed the cost of the flow alone, disregarding penalties. However, the penalty component subsequently imposed can be substantial. Apart from the WAF class where the objective gap is below 1%, for all other classes this gap is about 5% on average. Thus, using SH or HTC-MCNF rather than STC-MCNF may entail significant extra costs for a liner shipping operator that accounts for penalty functions.

All instances including WS and EUA could be solved quickly and in less than 6 s on average, meaning that a carefully designed CG method is extremely efficient to solve the STC-MCNF. The observed runtime on realistic LINER-LIB instances suggest that our approach scales well with the problem size and that, for all practical purposes, it is applicable to real-world cargo routing problems. Interestingly, despite being based on a standard shortest path pricing problem, SH runs in slightly more time for some instances compared to other methods. In fact, the CG for the standard MCNF requires more iterations to converge than that for the HTC/STC-MCNF.

It is a priori unclear if the single-commodity or one-to-all RCSP strategy performs best. Thus, we conducted additional experiments in which we solved the subproblems using each strategy and different combinations of fathoming rules. We found that employing the one-to-all RCSP strategy can further decrease runtime compared to the reported results. We relegate to Online Supplement D a detailed analysis of runtime, CG iterations, pricing strategies, and fathoming rules.

6.2. Impact of penalty coefficients on solution cost and gap

Next, we study in Fig. 6 how the total objective value and the optimality gaps of SH and HTC-MCNF vary with the penalty function, focusing on class EUA. Note that the x-axis scale is not linear but we use equidistant labels to improve visualization.

Varying the penalty slope for delays α_k (i.e. considering penalties **P1**) has no effect on HTC-MCNF, which is indifferent to penalties that apply to transit times beyond t_k^{MAX}. The HTC-MCNF cost is thus flat with respect to α_k as seen in Fig. 6(a). Increasing α_k, however, affects STC-MCNF as commodities flow faster to avoid large penalties; in extreme, the STC-MCNF solution converges to the solution of the HTC-MCNF. The cost of solutions obtained by SH is heavily affected by the choice of α_k, since SH neglects penalties first. Only afterwards, when potentially long routes have been selected, the respective cost which is proportional to α_k is added. Fig. 6(d) shows the corresponding optimality gap. It is interesting to see that α_k has opposite effects on the gap of SH and HTC-MCNF.

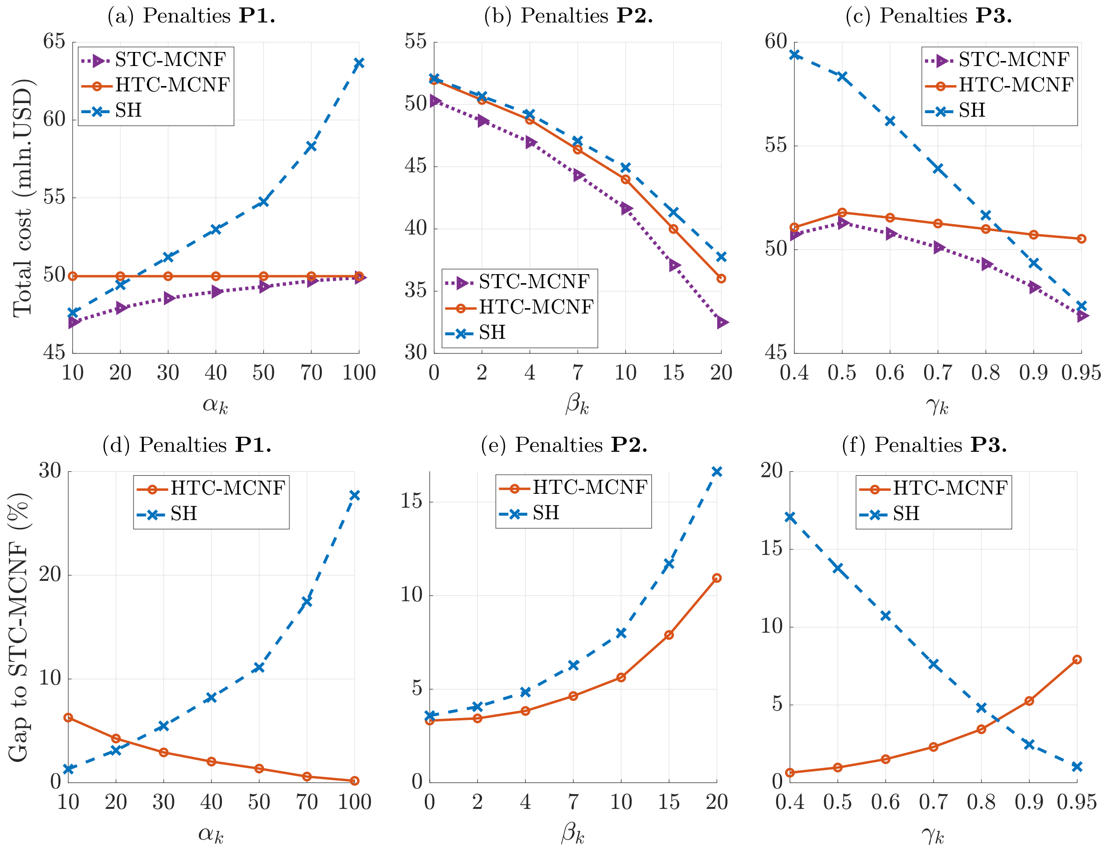


Fig. 6. Total costs and gaps to STC-MCNF at varying penalty function (EUA instances).

The objective function in all methods drops when increasing the discount slope β_k (penalties **P2**) as seen in Fig. 6(b). However, the associated gaps in SH and HTC-MCNF increase and reach significant levels above 10% or 15%, as shown in Fig. 6(e). This is because both these solutions are only "passively" subject to increasing discounts, whereas STC-MCNF exploits discounts by incorporating them into the optimization process, hence the objective decreases faster.

Penalties **P3** also affects costs and gaps as seen in Fig. 6(c) and (f). When γ_k is smaller (i.e. t_k^{MAX} is earlier in time), the gap is driven by the delay penalty and is therefore high in SH and low in HTC-MCNF. When γ_k is larger (t_k^{MAX} is later) the SH solution becomes closer to STC-MCNF and the gap is driven by the early arrival discounts. In summary, all three penalty coefficients play a role in the relative performance of the methods. In many cases, both SH and HTC-MCNF gaps are large, underscoring the importance of accounting for the penalties explicitly by solving the STC-MCNF.

6.3. Impact of penalty function on commodity transit time

In this section, we examine more in detail the flow transit time and how the penalty function affects it. Stating whether a commodity has traveled faster or slower in two solutions is unclear in principle because the same commodity might have used multiple, different paths in the two solutions. To establish a well-defined comparison, we introduce and use a flow-weighted average transit time:

$$\frac{\sum_{k\in\mathcal{K}}\sum_{p\in\mathcal{P}^{k,*}} t_p\, x_p^{k,*}}{\sum_{k\in\mathcal{K}} d_k} \tag{4}$$

where summations are taken over commodities $k \in \mathcal{K}$ and over the used paths $p \in \mathcal{P}^{k,*}$.

Fig. 7 shows for all instance classes how much the STC-MCNF flow-weighted average transit time (4) shifts up or down compared to the reference penalty when employing the coefficients from sets **P1**, **P2**, and **P3**. Commodities on average flow faster when the parameters α_k are increased (Fig. 7(a)), β_k are increased (Fig. 7(b)), γ_k are reduced (Fig. 7(c)), and they flow slower in the opposite cases. The transit time shift achievable using these penalties grows with the network size, also due to bigger networks such as WS and EUA covering larger geographical regions and hence including commodities that must be shipped to distant locations. Specifically, the flow-weighted transit time for classes WAF, MED, PAC, WS, and EUA (averaged over their 12 instances) is about 14, 13, 24, 31, and 36 days, respectively. Consequently, the shifts observed in Fig. 7 correspond to a significant range of roughly 4%–8% of the

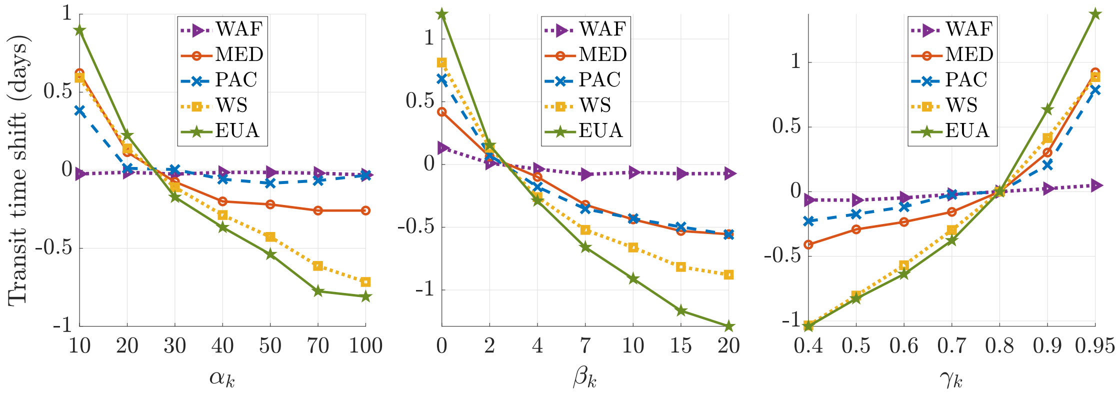


Fig. 7. Flow-weighted average transit time at varying penalty.

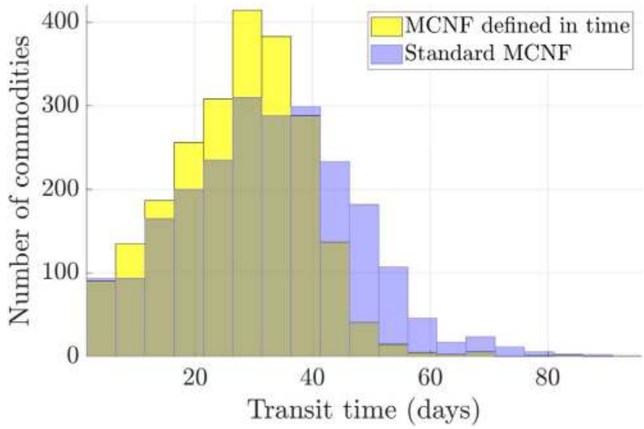


Fig. 8. Limiting commodity transit time distributions in one EUA instance.

transit time, except for WAF for which shifts are smaller. This analysis suggests that penalty functions are an effective lever to steer the flow towards different routing configurations that can be faster or slower. The liner shipping operator can design penalties to achieve a specific average transit time by modifying coefficients α_k, β_k, and γ_k.

Intuitively, if the penalty coefficients are set too high, the effort of the model will be mostly on minimizing the penalty and the original arc costs will be neglected. Therefore, a natural question that arises is how to make sure the penalty is "balanced", or, what is the risk of setting a too high penalty. Answering this question requires considering two extreme cases given by the standard MCNF and the MCNF defined in time (see, respectively, Steps 1 and 2 in Section 5.3). The latter model just minimizes the flow-weighted average transit time and hence can be seen as a limiting case when increasing the penalty function, providing the fastest cargo routing solution.

We compare in Fig. 8 the entire distribution of commodity flow-weighted transit times in these two extreme solutions for one EUA instance. As we can see from the figure, the histogram of the MCNF defined in time is visibly shifted to the left compared to the standard MCNF. In particular, the right tail of the distribution, denoting long transits of more than 40 days, becomes much thinner. Moreover, the mean of these distributions is 27.4 and 32.4 days, meaning that the maximum average speed-up achievable is 5 days. Knowing this difference is important for the liner shipping operator, as it quantifies the total managerial flexibility embedded in cargo routing that can be exploited via STC-MCNF regardless of the chosen penalty function and coefficients. Although this flexibility is bounded, our analysis shows it can be substantial (18% in this instance).

6.4. Managing the trade-off between cost and commodity transit time

Next, we investigate the key trade-off between cost of the flow and transit time faced by the liner shipping operator, and quantify the economic impact of implementing faster and slower commodity flows. For example, how much is the operator willing to pay to provide a one-day faster shipment to customers? Vice versa in order to cut operating costs by a certain amount, how much should the operator sacrifice in terms of transit time? In Fig. 9, we illustrate this trade-off for the EUA class on average over its 12

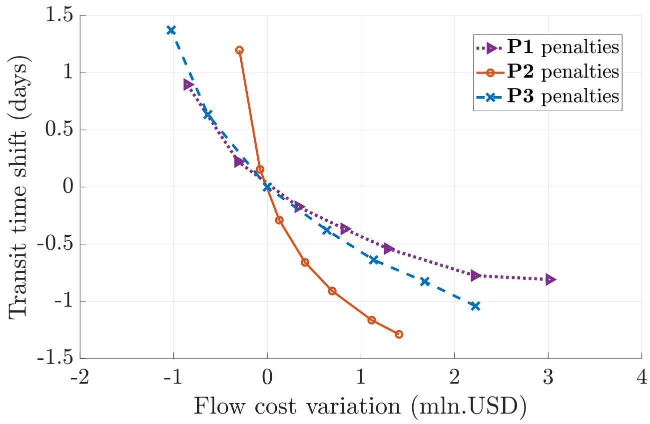


Fig. 9. Trade-offs between cost of the flow and average transit time.

instances. For each penalty set, we represent the transit time shift (days) and variation in cost (mln.USD) relative to the reference penalty coefficients. Note that for this trade-off to be meaningful it is important to consider the flow component of the cost alone and not the full objective function including penalties.

The graph reveals that the operator can manage the "speed-up" side of the trade-off (i.e., negative values on the y axis) more efficiently by increasing the bonus for early arrivals (penalties **P2**). In this case, a transit time reduction of 1.3 days (3.6%) can be achieved at a operational cost increase of 1.4 mln.USD (2.8%). These numbers reveal that reducing transit time is quite expensive; also because we consider an average speed-up of the entire flow (all commodities). We expect that liner operators may want to focus on a portion of the flow, e.g. the 5% or 10% of the commodities which are most time sensitive or belong to critical customers. Improving transit time for a subset of commodities would likely be associated with a much smaller cost increase. The "slow-down" side of the trade-off is instead managed more efficiently by penalties **P1** and **P3**. To save 1 mln.USD (2%), commodities would have to follow longer routes and reach destination 1.4 days (3.9%) later on average.

How should then a liner shipping operator position its business under the presented trade-off curve between cost and transit time? Reducing commodity transit time has multiple effects. On one hand, it results in an extra routing cost that our analysis helps quantifying. On the other hand, it generates direct savings such as decreased inventory costs for customers that could justify charging a higher rate, as well as indirect benefits for the firm including an increase in customer satisfaction and a decreased customer churn likelihood, which may be difficult to quantify. To understand if the speed-up is worthy, the operator should therefore compare both sides. (The situation is symmetric for the case of cargo slow down.) Consequently, managing this trade-off is non trivial from the operator's perspective as it involves balancing direct operational costs with hard-to-assess indirect consequences on reputation and customer satisfaction. Embedding penalty functions in cargo routing allows the operator to model these indirect effects, and the analysis in this paper may help in designing and parameterizing penalties based on their impact on the cargo flow.

6.5. Robustness and choice of the penalty function

Understanding the impact of penalty parameters on cost and transit time is important but the shape itself of the penalty function also plays a critical role (see the examples in Fig. 1). We thus consider another class of penalty functions with the goal to assess: (i) the robustness of our CG method, and (ii) how different penalties manage the trade-off between cost and commodity transit time. Specifically, we consider for each commodity a staircase penalty function with three *regimes*: a discount for low transit times, a zero penalty for intermediate transit times, and a delay cost for high transit times. Similarly to the piecewise linear penalty functions (hereafter simply linear penalties), we tested staircase penalties under a wide range of parameter variations involving the step values and the transit times splitting the three regimes. Notice that even simple staircase penalties with two or three steps require the use of the STC-MCNF. Relying on the HTC-MCNF by varying the hard limits on maximum transit time is not viable because penalties are commodity-specific and the number of scenarios to explore thus grows exponentially with $|\mathcal{K}|$.

Solving the STC-MCNF with staircase penalties, we found that the running time of CG was almost identical to the case of linear penalties, not exceeding 4–5 s for the largest EUA instances. This suggests that the CG method is robust to the functional choice of the penalty as its performance is hardly affected. We then computed the trade-offs between flow cost (excluding penalty) and transit time, and report their averages across EUA instances in Fig. 10. For comparison, we also show the analogous trade-offs from linear penalties discussed in Section 6.4. For clarity, we report here variations in cost and transit time relative to the standard MCNF solution, with coordinates (0,0). The best solutions in this graphs are those that can achieve large reductions in average transit time for small increase in routing costs, i.e., those located closer to the bottom-left corner.

The figure shows that linear penalties are significantly more efficient than staircase penalties to handle this trade-off. For a cost increase of 1–2 mln.USD, linear penalties can decrease the average commodity transit time by more than 4 days compared to the standard MCNF (zero penalty), while staircase penalties never meet the 2-day bar. The reason behind this different efficiency

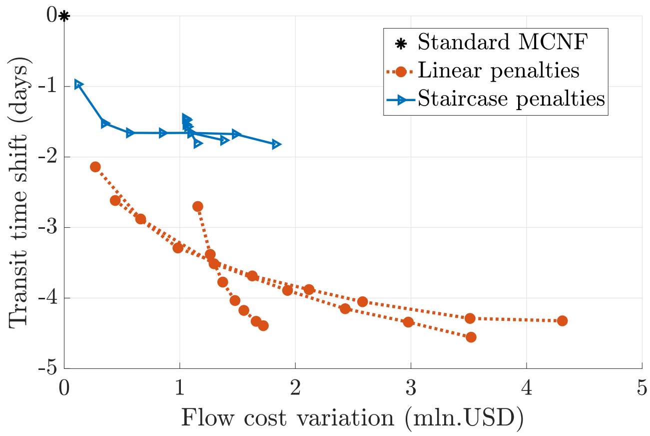


Fig. 10. Shifts in cost of the flow and average transit time under linear and staircase penalties. The values are relative to the standard MCNF solution in (0,0), i.e., at the top-left of the graph.

is that staircase penalties consider two paths within the same regime equal, even though one arrives later. In other words, these penalties do not incentivize commodities to flow faster within a given regime but only to "jump" between one regime to another (e.g., from zero penalty to reward). Instead, linear penalties push commodities to speed up even when the resulting improvement is small, which on average across commodities lead to the large difference observed in Fig. 10. These findings support the use of linear penalties to achieve more economical reductions of commodity transit time.

Despite linear penalties being more efficient, a staircase penalty with two or three regimes is simpler and more intuitive. For this reason, such a function is popular in retail e-commerce applications in which the customer is directly faced with the options of fast shipping at an extra charge, standard shipping, and in some cases also a "no-rush" shipping associated with a reward/bonus (Amazon, 2021). In this situation, it makes sense to use a staircase penalty that is easily understood by customers, while linear penalties would be impractical to enforce.

In liner shipping, however, penalties usually do not reflect actual payments to the clients, which traditionally do not get compensated for late arrivals. Nevertheless, there are intangible costs due to customer dissatisfaction and lost reputation, so internally the company can define linear penalty functions as a way of quantifying those intangible costs caused by late arrivals. Linear penalties seem more appropriate for a liner shipping operator, also because they more accurately reflect inventory costs (including e.g. cooling expenses during the transportation) that increase linearly with time.

7. Conclusion

Motivated by a major liner shipping operator, we studied a variant of the classical MCNF that incorporates soft constraints on commodity transit times. The STC-MCNF indeed addresses the shortcomings of existing and less flexible approaches based on hard transit time constraints, e.g., that paths with a tiny delay with respect to a target commodity transit time would be discarded while the industry may be willing to accept such solutions. We derived properties of the STC-MCNF that highlight the differences with both the HTC-MCNF and the sequential heuristic SH, and hence further motivate studying this problem. We then adapted a column generation method and the fathoming rules of the pricing problem, a resource-constrained shortest path variant, to solve the STC-MCNF very efficiently even for large networks. Using realistic liner shipping instances based on the LINER-LIB data, we examined solution costs, optimality gaps, and commodity transit times under varying penalty functions. The key findings from our numerical experiments are the following:

1. HTC-MCNF and SH that do not consider penalty functions explicitly may result in large optimality gaps compared to STC-MCNF, which instead optimizes flows accounting for penalties;
2. Penalty functions can be used as a lever to steer the flow towards slower/faster configurations;
3. Liner shipping operators face a trade-off between cost of the flow and commodity transit time, managing which is challenging as it entails comparing tangible direct costs with indirect costs and implications for the company. Our study helps quantifying relevant components of this trade-off and how they vary depending on the penalty function parameterization.
4. We provide support for considering piecewise linear penalties in liner shipping networks, which are more efficient in reducing transportation time than simpler staircase penalties popular in retail e-commerce applications.

Further research directions include studying two extensions of this problem. First, the STC-MCNF could be incorporated in the liner shipping network design problem. In that case, the penalties would play a role not only in routing the containers but also in designing the network, i.e. selecting the liner shipping services to use. Second, it would be interesting to include arc transit times as decision variables (i.e. optimizing the speed of vessels) which would further increase the flexibility of the model and make target transit time reductions cheaper to obtain for the operator.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Alessio Trivella and Francesco Corman are supported by the Swiss National Science Foundation, Switzerland under the research project DADA, grant no. 181210. David Koza and David Pisinger are supported by the Innovation Fund Denmark under the research project GREENSHIP, grant no. 1313-00005B. All authors approved the version of the manuscript to be published.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.tre.2021.102342.

References

Agarwal, R., Ergun, Ö., 2008. Ship scheduling and network design for cargo routing in liner shipping. Transp. Sci. 42 (2), 175–196.
Ahuja, R., Magnanti, T., Orlin, J., 1993. Network Flows: Theory, Algorithms, and Applications. Prentice Hall, Upper Saddle River, NJ, USA.
Amazon, 2021. Don't need it urgently? Earn a digital reward. URL https://www.amazon.com/b?ie=UTF8&node=9433645011, Accessed 3 March 2021.
Azadi Moghaddam Arani, A., Jolai, F., Nasiri, M., 2019. A multi-commodity network flow model for railway capacity optimization in case of line blockage. Int. J. Rail Transp. 7 (4), 297–320.
Barnhart, C., Hane, C., Vance, P., 2000. Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. Oper. Res. 48 (2), 318–326.
Bloomberg, 2020. The 'mad rush' for medical gear triples air cargo rates. URL https://www.bloomberg.com/news/articles/2020-04-16/supply-chains-latest-virus-sees-air-freight-costs-soar.
Boland, N., Dethridge, J., Dumitrescu, I., 2006. Accelerated label setting algorithms for the elementary resource constrained shortest path problem. Oper. Res. Lett. 34 (1), 58–68.
Brouer, B., Alvarez, J., Plum, C., Pisinger, D., Sigurd, M., 2012. LINER-LIB. URL http://www.linerlib.org/.
Brouer, B., Alvarez, J., Plum, C., Pisinger, D., Sigurd, M., 2013a. A base integer programming model and benchmark suite for liner-shipping network design. Transp. Sci. 48 (2), 281–312.
Brouer, B., Dirksen, J., Pisinger, D., Plum, C., Vaaben, B., 2013b. The vessel schedule recovery problem (VSRP)–a MIP model for handling disruptions in liner shipping. European J. Oper. Res. 224 (2), 362–374.
Brouer, B., Pisinger, D., Spoorendonk, S., 2011. Liner shipping cargo allocation with repositioning of empty containers. INFOR: Inform. Syst. Oper. Res. 49 (2), 109–124.
Christiansen, M., Hellsten, E., Pisinger, D., Sacramento, D., Vilhelmsen, C., 2019. Liner shipping network design. European J. Oper. Res. In Press, pp, 1–20.
Desaulniers, G., Desrosiers, J., Solomon, M., 2005. Column Generation. Springer, New York, NY, USA.
Duque, D., Lozano, L., Medaglia, A., 2015. An exact method for the biobjective shortest path problem for large-scale road networks. European J. Oper. Res. 242 (3), 788–797.
Enzi, M., Parragh, S., Pisinger, D., Prandtstetter, M., 2021. Modeling and solving the multimodal car-and ride-sharing problem. European J. Oper. Res. 293 (1), 290–303.
Fragkos, I., Cordeau, J., Jans, R., 2017. The multi-period multi-commodity network design problem. Technical Report, CIRRELT, Montreal, QC, Canada.
Garey, M., Johnson, D., 1990. Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA.
Gelareh, S., Nickel, S., Pisinger, D., 2010. Liner shipping hub network design in a competitive environment. Transp. Res. Part E 46 (6), 991–1004.
Guericke, S., Tierney, K., 2015. Liner shipping cargo allocation with service levels and speed optimization. Transp. Res. Part E 84, 40–60.
Hassin, R., 1992. Approximation schemes for the restricted shortest path problem. Math. Oper. Res. 17 (1), 36–42.
Holmberg, K., Yuan, D., 2003. A multicommodity network-flow problem with side constraints on paths solved by column generation. INFORMS J. Comput. 15 (1), 42–57.
Irnich, S., Desaulniers, G., 2005. Shortest path problems with resource constraints. In: Column Generation. Springer, New York, NY, USA, pp. 33–65.
Karsten, C., Pisinger, D., Ropke, S., Brouer, B., 2015. The time constrained multi-commodity network flow problem and its application to liner shipping network design. Transp. Res. Part E 76 (1), 122–138.
Koza, D., 2017. Models and Methods for the Design and Support of Liner Shipping Networks (PhD thesis). DTU Management Engineering, Kgs. Lyngby, Denmark.
Koza, D., 2019. Liner shipping service scheduling and cargo allocation. European J. Oper. Res. 275 (3), 897–915.
Koza, D., Desaulniers, G., Ropke, S., 2020. Integrated liner shipping network design and scheduling. Transp. Sci. 54 (2), 512–533.
Li, H., Zhang, T., Zhang, Y., Wang, K., Li, J., 2017. A maximum flow algorithm based on storage time aggregated graph for delay-tolerant networks. Ad Hoc Netw. 59 (1), 63–70.
Lozano, L., Medaglia, A., 2013. On an exact method for the constrained shortest path problem. Comput. Oper. Res. 40 (1), 378–384.
Melchiori, A., Sgalambro, A., 2020. A branch and price algorithm to solve the quickest multicommodity k-splittable flow problem. European J. Oper. Res. 282 (1), 846–857.
Meng, Q., Wang, S., Andersson, H., Thun, K., 2013. Containership routing and scheduling in liner shipping: overview and future research directions. Transp. Sci. 48 (2), 265–280.
Minoux, M., 2006. Multicommodity network flow models and algorithms in telecommunications. In: Handbook of Optimization in Telecommunications. Springer, Boston, MA, USA, pp. 163–184.
Notteboom, T., 2006. The time factor in liner shipping services. Maritime Econ. Logist. 8 (1), 19–39.
Othelius, J., Wemmert, U., 2014. Analysis of customer needs and service quality at a liner shipping company (Masters thesis). Chalmers University of Technology, Göteborg, Sweden.
Pisinger, D., 2016. Liner shipping network design - a new decomposition. In: EURO-2016 28th European Conference on Operational Research, Poznan, 3-6 July, 2016.
Pugliese, L., Guerriero, F., 2013. A reference point approach for the resource constrained shortest path problems. Transp. Sci. 47 (2), 247–265.
Reinhardt, L., Pisinger, D., 2011. Multi-objective and multi-constrained non-additive shortest path problems. Comput. Oper. Res. 38 (3), 605–616.

Reinhardt, L.B., Pisinger, D., Sigurd, M., Ahmt, J., 2020. Speed optimizations for liner networks with business constraints. European J. Oper. Res. 285 (3), 1127–1140.
Sedeño-Noda, A., Colebrook, M., 2019. A biobjective dijkstra algorithm. European J. Oper. Res. 276 (1), 106–118.
Venturini, G., Iris, Ç., Kontovas, C., Larsen, A., 2017. The multi-port berth allocation problem with speed optimization and emission considerations. Transp. Res. Part D 54, 142–159.
Wang, P., Li, H., Zhang, T., Zhang, S., Shi, K., 2019. A novel dynamic multi-source multi-sink flow algorithm over the satellite networks. In: International Conference on Wireless and Satellite Systems. Springer, Cham, Switzerland, pp. 303–311.
Wang, S., Meng, Q., 2012. Robust schedule design for liner shipping services. Transp. Res. Part E 48 (6), 1093–1106.
Wang, S., Meng, Q., 2014. Liner shipping network design with deadlines. Comput. Oper. Res. 41, 140–149.
Wang, S., Meng, Q., Lee, C., 2016. Liner container assignment model with transit-time-sensitive container shipment demand and its applications. Transp. Res. B 90 (1), 135–155.
Wang, S., Meng, Q., Liu, Z., 2013a. Containership scheduling with transit-time-sensitive container shipment demand. Transp. Res. B 54 (1), 68–83.
Wang, S., Meng, Q., Sun, Z., 2013b. Container routing in liner shipping. Transp. Res. Part E 49 (1), 1–7.
Xia, J., Li, K., Ma, H., Xu, Z., 2015. Joint planning of fleet deployment, speed optimization, and cargo allocation for liner shipping. Transp. Sci. 49 (4), 922–938.
Zhang, T., Li, H., Li, J., Zhang, S., Shen, H., 2018. A dynamic combined flow algorithm for the two-commodity max-flow problem over delay-tolerant networks. IEEE Trans. Wireless Commun. 17 (12), 7879–7893.