Incorporating Holding Costs in Continuous-Time Service Network Design: New Model, Relaxation, and Exact Algorithm

Shengnan Shu
Department of Logistics and Maritime Studies, the Hong Kong Polytechnic University, shengnan.shu@connect.polyu.hk

Zhou Xu
Department of Logistics and Maritime Studies, the Hong Kong Polytechnic University, zhou.xu@polyu.edu.hk

Roberto Baldacci
Division of Engineering Management and Decision Sciences, College of Science and Engineering, Hamad Bin Khalifa University, Qatar Foundation, Doha, Qatar, rbaldacci@hbku.edu.qa

The continuous-time service network design problem (CTSNDP) occurs widely in practice. It aims to minimize the total operational cost by optimizing the schedules of transportation services and the routes of shipments for dispatching, which can occur at any time point along a continuous planning horizon. In order to be cost effective, shipments often wait to be consolidated, which incurs a holding cost. Despite its importance, the holding cost has not been taken into account in existing exact solution methods for the CTSNDP, since introducing it significantly complicates the problem and makes the solution development very challenging. To tackle this challenge, we develop a new dynamic discretization discovery algorithm, which can solve the CTSNDP with holding cost to exactly optimum. The algorithm is based on a novel relaxation model and several new optimization techniques. Results from extensive computational experiments validate the efficiency and effectiveness of the new algorithm, and also demonstrate the benefits that can be gained by taking into account holding costs in solving the CTSNDP. In particular, we show that the significance of the benefits depends on the connectivity of the underlying physical network, and on the flexibility of the shipments' time requirements.

Key words: Transportation : Network, Networks-graphs : Multi-commodity, Programming : Integer : Algorithms

History: Jan 24, 2023

1. Introduction

Service network design problems (Crainic 2000) are common and important problems in transportation, telecommunications, logistics, and production–distribution systems. In the freight transportation industry, the less-than-truckload (LTL) motor carriers are typical examples of such systems, where an intensive use of freight consolidation operations are performed to save on transportation costs (Wieberneit 2008).

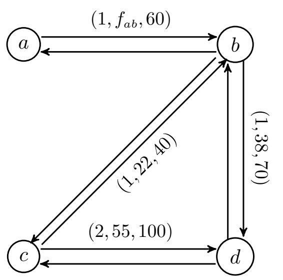
The service network design problem considered in this paper, referred to as the SNDP, can be described as follows. A network $\mathcal{D} = (\mathcal{N}, \mathcal{A})$ is given with terminal or node set $\mathcal{N}$ and arc set $\mathcal{A}$. Let $\mathcal{K}$ be a set of commodities, each commodity $k \in \mathcal{K}$ has an origin $o^k \in \mathcal{N}$, a destination $d^k \in \mathcal{N}$, and a transportation demand $q^k \in \mathbb{N}_{>0}$ that must be delivered to the destination from the origin. While flowing along an arc (i,j), a commodity consumes some of the arc capacity; the capacity is obtained by installing on some of the arcs any number of *links*. In network $\mathcal{D}$, also referred as a *flat* network, each arc $(i,j) \in \mathcal{A}$ is associated with the following four attributes: (i) a travel time $\tau_{ij} \in \mathbb{N}_{>0}$; (ii) a per-unit-of-flow cost $c_{ij}^k \in \mathbb{R}_{>0}$ for each commodity $k \in \mathcal{K}$; (iii) a fixed cost $f_{ij} \in \mathbb{R}_{>0}$; and (iv) a capacity $u_{ij} \in \mathbb{N}_{>0}$. Installing one link on arc (i,j) provides a capacity u_{ij} at a cost f_{ij}. With each commodity $k \in \mathcal{K}$ is also associated an earliest available time $e^k \in \mathbb{N}_{\geq 0}$ at the origin and a latest arrival time $l^k \in \mathbb{N}_{\geq 0}$ at the destination. We consider the *unsplittable* (or unbifurcated) variant of the problem, where the flow of each commodity is required to follow one route between the origin and the destination, as also considered by Boland et al. (2017) and Marshall et al. (2021).

The SNDP consists of minimizing the sum of all costs (both fixed and flow costs), while at the same time satisfying demand requirements, as well as capacity and time constraints. The SNDP is known to be strongly $\mathcal{NP}$–hard (Ghamlouche et al. 2003), and various extensions of the SNDP have been studied in the transportation and telecommunications fields (Gendron et al. 1999, Frangioni and Gendron 2009).

In the SNDP, the decisions are made as to the schedule of the services, this schedule specifying timing information for each possible occurrence of a service during a given time period (i.e., departure and arrival times at the origins, intermediate stops, and destinations). A common technique adopted in the literature for modeling the temporal component is *discretization* (Jarrah et al. 2009, Andersen et al. 2011, Erera et al. 2013, Crainic et al. 2014), where the planning horizon is discretized, and the problem is modeled on a *time-expanded network*. In the network, nodes represent locations in time and space, while arcs or links represent either physical movements between locations or just movements in time at one location. More precisely, the arcs on the network are classified into *dispatch* or *service* arcs and *holding* arcs. A service arc corresponds to the transportation between two locations, and the difference between the periods of these locations is the time elapsed during the transportation activity, whereas a holding arc is directed from one period to another for the same location and represents only time-wise movement. The granularity of the time discretization has an impact on both the computational tractability and the quality of the solutions obtained, and studies have been presented that accurately capture the consolidation opportunities as a Continuous-Time SNDP (CTSNDP) (Boland et al. 2017, Marshall et al. 2021).

1.1. The SNDP with holding costs

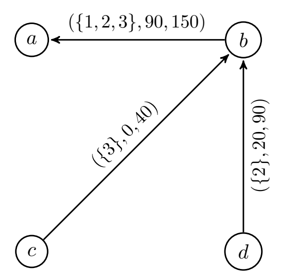
For many practical applications of the SNDP, *holding costs* have a significant impact on the service and consolidation decisions (Tyan et al. 2003, Bookbinder and Higginson 2002, Rudi et al. 2016, Hu et al. 2018). These costs are also called *consolidation penalty costs* (Ülkü 2009*b*), and can be facility-specific and/or commodity-specific (Hu et al. 2018, Ulku 2009*a*, Rudi et al. 2016). In



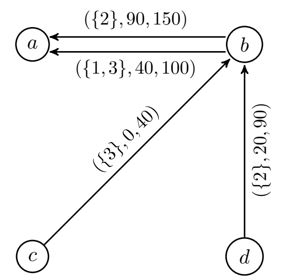
(a) Network $\mathcal{D}$: (c, f, τ) on arcs

k	o^k	d^k	q^k	e^k	l^k
1	b	a	25	0	160
2	d	a	30	20	180
3	c	a	40	0	180

(b) Instance data



(c) Without holding costs



(d) With holding costs

Figure 1 Examples of SNDP solutions (Figures (c) and (d): $(\{commodities\}, dep.time, arr.time)$ **on arcs)**

the literature, holding costs are classified as *in-transit* and *in-storage*, the in-transit holding cost usually being lower than the in-storage cost. In the context of time-expanded networks, these costs are generally modeled by properly defining the costs associated with the service and holding arcs of the network.

Motivated by the importance of the continuous variant of the SNDP and of the holding costs, in this paper we consider the CTSNDP with both *in-transit* and *in-storage* holding costs or simply *holding costs* (CTSNDP-HC). In the following, *in-transit* holding costs are modeled by means of costs c_{ij}^k, whereas to model *in-storage* holding costs, we associate with each commodity $k \in \mathcal{K}$ and node $i \in \mathcal{N}$ a per-unit-of-demand-and-time (holding) cost $h_i^k \in \mathbb{R}_{\geq 0}$. Hereafter, we also use the term *holding costs* to refer to both the *in-transit* and *in-storage* holding costs.

To highlight the importance of considering the holding costs in the SNDP, Figure 1 gives a simple example of a SNDP instance with nonzero holding costs. The example involves three commodities. Figure 1-(a) depicts the underlying network $\mathcal{D}$ where relevant flow and fixed costs and travel times are reported close to each arc. In the example, arcs (i, j) and (j, i) share the same data. Figure 1-(b) gives the different parameters associated with the three commodities. In addition, each arc capacity is assumed to be greater than the total demand of the commodities, i.e., to be 100.

Figure 1-(c) illustrates an optimal solution for the SNDP by disregarding the holding costs. The figure reports on each arc (i, j) the set of commodities consolidated, the departure time from node i, and the arrival time to node j, represented by a triplet $(\{commodities\}, dep.time, arr.time)$. The solution of Figure 1-(c) shows flow and fixed costs equal to 165 (=1 $\times$ 40 (c, b) + 1 $\times$ 30 (d, b) + 1 $\times$ (25+30+40) (b, a)) and $60 + f_{ba}$ (=22 (c, b) + 38 (d, b) + f_{ba}), respectively. The three commodities are consolidated on arc (b, a), where commodities 1 and 3 wait 90 and 50 time units before being consolidated with commodity 2, which arrives at node b at time 90.

We now assume that the in-storage per-unit-of-demand-and-time holding cost for each commodity at nodes b, c and d is equal to η (i.e., $h_b^k = h_c^k = h_d^k = \eta$, $k = 1, 2, 3$) whereas the holding cost at the destination node a is assumed to be equal to zero. Figure 1-(d) illustrates an alternative solution for the SNDP. In the solution, commodity 2 is routed directly from its origin c to the destination a without consolidation operations whereas commodities 1 and 3 are consolidated on arc (b,a). Solution 1-(d) shows flow and fixed costs equal to 165 (=1 × 40 (c,a) + 1 × 30 (d,b) + 1 × (25+30+40) (b,a)) and $60+2\times f_{ba}$ (=22 (c,a) + 38 $(d,b)+2\times f_{ba}(b,a)$), respectively. The holding cost of commodity 1 at node b is equal to $40\times\eta\times 25$. By considering the holding cost of solution (c) which is equal to $(90\times\eta\times 25)+(50\times\eta\times 40)$, the difference between the total costs of solutions (c) and (d) is therefore equal to $\mathit{diff} = 50\times\eta\times 25+50\times\eta\times 40-f_{ba}$. If η is set equal to 0.01 and $f_{ba}<32.5$, $\mathit{diff}>0$, i.e., solution (c) is a feasible but not optimal solution for the SNDP with consideration of the holding costs. If, for example, $f_{ba}=10$ and $\eta=0.01$, an optimal solution for the SNDP with consideration of the holding costs is given by solution (d) with a cost equal to 255, showing a saving of about 8% with respect to the cost 277.5 of solution (c).

1.2. Discretized versus continuous-time models

A time-expanded network provides a useful way of modeling the SNDP, but the corresponding time-index (TI) model (see, for example, Fleischer and Skutella 2007, Groß and Skutella 2012) requires a discretization of time known to be fine enough to provide a correct model for the continuous time, i.e., to show that its optimal solution cost is *continuous-time* optimal. What is more, choosing an appropriate time discretization can be challenging. On the one hand, a fine discretization results in good approximations to the continuous-time problem, but at the expense of a large (and generally intractable) TI model. On the other hand, a coarse discretization is more computationally practical, but the *price of discretization* (i.e., the loss of solution quality) can be very high (Boland et al. 2018).

In this context, it is beneficial to investigate *complete* TI models based on a *complete* discretization of time, i.e., a discretization of time known to be fine enough to provide a correct model for the continuous time. As shown by Boland and Savelsbergh (2019), the existence of a complete TI model is not straightforward.

1.3. Contributions of this paper

In this paper, we describe an exact algorithm for the CTSNDP-HC. The method is based on the Dynamic Discretization Discovery (DDD) solution framework proposed by Boland et al. (2017) to solve the CTSNDP. The DDD uses successive approximations of a TI model in order to obtain the optimal continuous-time solution, and its correctness relies on the existence of a complete TI model for the problem. The main ingredients of a DDD are:

- A valid *relaxation* of a complete TI model based on a partial discretization.
- A *primal* heuristic that uses the solution provided by the relaxation to compute a valid upper bound on the optimal value of the complete TI model. If the cost of the primal solution is equal to the solution cost of the relaxation, then the primal solution is proved to be optimal.

• A *refinement strategy* that, given a solution of the relaxation, refines the current partial discretization so that a new valid relaxation for the complete TI model is derived, the current solution of the relaxation is no longer feasible for the new relaxation model, and the convergence of the algorithm is guaranteed.

The DDD algorithm proposed by Boland et al. (2017) for the CTSNDP relies strongly on the assumption that freight can be held at a location at no cost, that is, in-storage holding costs are equal to zero, and cannot be used to solve the CTSNDP-HC. Our distinct contributions in this paper are as follows:

• We prove the existence of a complete TI model for the CTSNDP-HC.

• We derive a new relaxation of the complete TI model based on a mixed-integer linear programming (MIP) model.

• Based on the complete TI model and its new relaxation, we develop a new DDD algorithm with a new upper bound heuristic and a new refinement strategy to solve the CTSNDP-HC.

• We validate the efficiency and effectiveness of the new algorithm via extensive computational experiments. The computational results also show the benefits that can be gained by incorporating holding costs. The significance of the benefits turns out to depend upon the connectivity of the underlying physical network and the flexibility of the shipments' time requirements.

Our work not only enriches the optimization techniques for the CTSNDP and CTSNDP-HC, but also enhances the DDD algorithm and extends its applications. It is potentially useful in facilitating future study on solving various transportation network design problems with holding costs.

The remainder of this paper is organized as follows. After a literature revew in §2, we prove the existence of a complete TI model for the CTSNDP-HC in §3. We present our newly developed DDD algorithm for the CTSNDP-HC in §4. In particular, we introduce the new relaxation in §4.1, and illustrate the algorithm details, including a primal heuristic and a refinement strategy, in §4.3 and §4.4. We then present the computational studies in §5. The paper is concluded in §6 with a discussion on future research directions.

2. Related works

Service network design problems have been widely studied in the literature since the 1990s (Crainic and Rousseau 1986, Farvolden and Powell 1994) due to the wide range of applications they cover (Crainic 2000, Wieberneit 2008). An early classification distinguishes between *static* and *dynamic* service network design problems, where in the dynamic variant the timing aspects of the service routes are highlighted. For a review of the static variants and associated applications, the reader is referred to Crainic (2000) and Wieberneit (2008). Below, we focus on the works closely related to the SNDP and to the solution approaches based on discretization methods (§2.1). Further, we refer to the literature on service network design problems with the objective of capturing consolidation costs such as in-storage holding costs (§2.2).

2.1. Discretization and dynamic discretization discovery methods

The literature shows different discretization methods aimed at deriving relaxations of TI models for routing and scheduling problems with time constraints. Wang and Regan (2002) and Wang and Regan (2009) proposed relaxations of TI formulations for a vehicle routing problem and for the Traveling Salesman Problem with Time Windows (TSPTW), respectively. The relaxations are obtained by partitioning the time windows into a collection of nonoverlapping intervals and by defining variables for these intervals. The resulting relaxations are then exploited to devise strong cutting planes that are embedded in a branch-and-cut framework. A similar relaxation, called *time bucket relaxation*, was investigated by Dash et al. (2012) to solve the TSPTW by a branch-and-cut algorithm that also makes use of valid inequalities derived from the bucket formulation.

Boland et al. (2017) introduced the DDD to solve the CTSNDP. The method solves a sequence of MIPs defined on a subset of times (i.e., a partial discretization), with variables indexed by times in the subset, that provides lower bounds on the optimal continuous-time value. At each iteration of the method, new times are discovered and used to refine the partial discretization. Once the right subset of times is discovered, the resulting MIP model yields the continuous-time optimal value. As highlighted by Boland et al. (2017), the refinement strategies of Wang and Regan (2009) and Dash et al. (2012) employ a DDD as a preprocessing scheme, rather than a dynamic nonuniform scheme. Further, Boland et al. (2017) also focus on the size of the partially time-expanded network by keeping the number of time points in the network to a minimum. The recent work of Marshall et al. (2021) further extends that of Boland et al. (2017) by modeling the discretization in terms of time intervals instead of time points. This new discretization leads to more effective and efficient DDD algorithms. The algorithm of Marshall et al. (2021) can handle larger instances involving up to 30 nodes, 685 arcs, and 400 commodities, and can generate high-quality solutions more quickly than that of Boland et al. (2017).

Solution methods based on the DDD solution framework for service network design problems were also investigated by Hewitt (2019) and Medina et al. (2019). Hewitt (2019) considered variants of the service network design problem encountered in the LTL freight transportation industry. They proposed multiple enhancements to the DDD algorithmic framework based on inequalities and symmetry-breaking branching rules. Medina et al. (2019) introduced an optimization problem that integrates long-haul and local transportation planning decisions. The authors proposed both a route-based and an arc-based formulation for the problem that are solved by means of a DDD algorithm. Recently, Hewitt (2022) investigated the scheduled service network design problem that can support planning the transportation operations of consolidation carriers given shipment-level service commitments regarding available and due times. In particular, Hewitt (2022) considered flexibility on the available and due times to minimize total transportation and handling costs. The author proposed an adaptation of DDD to solve the problem and studied the savings potential of leveraging flexibility. Other applications of the DDD solution framework can be found in Vu et al. (2020), whereas for further perspectives on various aspects of time-dependent models and the DDD the reader is referred to Boland and Savelsbergh (2019).

All the aforementioned works disregard holding costs. In particular, as we will show later, the correctness of the DDD approach proposed by Boland et al. (2017) strongly relies on the assumption that the in-storage holding costs are equal to zero.

2.2. Handling consolidation costs

Several works have highlighted the importance of considering consolidation costs in service network design. Ülkü (2009*b*) addressed holding costs as consolidation penalty costs, and presented three shipment consolidation policies, namely, time, quantity, and hybrid policies. Pedersen et al. (2009) focused on a generic model for transportation service network design with asset management considerations. The authors modeled asset positioning and utilization through constraints on asset availability at terminals, with the consideration of in-storage holding costs. The problem was formulated by means of an arc-based model, and a tabu search metaheuristic was used for its solution. Rudi et al. (2016) investigated a capacitated multi-commodity network flow model for the planning of intermodal transportation services considering carbon emissions and in-transit holding costs. The application of the model on a set of industry data investigated the interrelations between the decision criteria regarding greenhouse gas emissions, cost, and time, as well as the influence of inventory holding costs. Jarrah et al. (2009) and Erera et al. (2013) investigated real-world service network design problems faced by LTL freight transportation carriers. Both of the works considered handling costs at intermediate terminals. Jarrah et al. (2009) described an IP formulation capturing the different LTL requirements that is solved using a slope scaling and load-planning tree generation method. Erera et al. (2013) presented integer linear programming (IP) models and a matheuristic solution approach for large-scale instances that result in practical applications. Additional works analyzing the trade-off between transportation and holding costs can be found in Bookbinder and Higginson (2002), Tyan et al. (2003), Ulku (2009*a*) and Hu et al. (2018).

Holding costs for the SNDP and its variants formulated using TI models have been considered by several works, such as Andersen et al. (2009*b*,*a*), Pedersen et al. (2009), and Crainic et al. (2018). However, due to the approximation introduced by the discretization, the solution methods proposed in these works cannot guarantee the optimality of the solutions obtained. To the best of our knowledge, no exact algorithm has been proposed for the CTSNDP-HC, and the related literature is quite scarce. A continuous SNDP with vehicle asset management was investigated by Hosseininasab (2015), where vehicle waiting and holding costs were also considered. Belieres (2019) considered tactical transportation planning in a multi-product supply chain inspired by the collaboration between a third-party logistics company and a restaurant chain. The problem was formulated using a TI model with holding costs, and was solved by means of a hybrid matheuristic based on the DDD. The author observed that the DDD method proposed by Boland et al. (2017) cannot be used in the presence of in-storage holding costs. The algorithm was tested on real-world instances, and the results show that refining the granularity of the time discretization generates substantial savings in terms of holding costs.

3. Modeling the CTSNDP-HC on a finite time-expanded network

In this section, we first describe the structure of feasible CTSNDP-HC solutions. We then describe a TI model for the CTSNDP-HC and we show the existence of a complete TI model.

3.1. Representing feasible solutions

A *path* $P^k = (v_1^k, v_2^k, ..., v_{\eta^k+1}^k)$ for a commodity $k \in \mathcal{K}$ is a not necessarily elementary path in $\mathcal{D}$ starting from node $v_1^k = o^k$ and ending at node $v_{\eta^k+1}^k = d^k$. Associated with path P^k is also the sequence $(a_1^k, a_2^k, ..., a_{\eta^k}^k)$ of arcs traversed by the path such that $a_n^k = (v_n^k, v_{n+1}^k) \in \mathcal{A}$ for $n = 1, 2, ..., \eta^k$; in the following, the two representations of path P^k are used interchangeably. Given a set of departures times $t^k = (t_1^k, t_2^k, ..., t_{\eta^k}^k)$ associated with the nodes of the path, path P^k is k-feasible, and we denote it with the pair $\mathcal{W}^k = (P^k, t^k)$, if values t_n^k, $n = 1, \dots, \eta^k$, satisfy the following system of inequalities:

$$t_n^k \geq e^k, \qquad n = 1, \tag{1a}$$

$$t_n^k \geq t_{n-1}^k + \tau_{a_{n-1}^k}, \quad n = 2, \dots, \eta^k, \tag{1b}$$

$$t_n^k + \tau_{a_n^k} \leq l^k, \qquad n = \eta^k. \tag{1c}$$

Inequalities (1a) and (1c) impose that the departure and arrival times at the origin and destination are within the required time limits e^k and l^k, respectively, where the term $t_n^k + \tau_{a_n^k}$ coincides also with the departure time at the destination d^k. Inequalities (1b) impose feasible departure times at the intermediate nodes of the path.

We also associate with each arc $a_n^k \in P^k$, $n = 1, 2, \dots, \eta^k$, a departure time t_n^k. A *feasible* solution $\mathcal{W} = \{\mathcal{W}^k\}_{k \in \mathcal{K}}$ of the CTSNDP-HC is a collection of $|\mathcal{K}|$ paths, one feasible path for each commodity. We assume that for each commodity $k \in \mathcal{K}$, the difference $(l^k - e^k)$ of its latest arrival time l^k at the destination and available time e^k at the origin is not smaller than the length of the shortest-time path from o^k to d^k in the flat network $\mathcal{D}$. This assumption is sufficient to ensure the existence of a feasible solution to the CTSNDP-HC.

Given a k-feasible timed path $\mathcal{W}^k$, the *holding plan* of the path is defined as the set of the waiting times δ_n^k, $n=1\ldots,\eta^k+1$, at the different nodes of the path that can be computed as follows:

$$\delta_n^k = \begin{cases} t_n^k - e^k, & n=1, \\ t_n^k - (t_{n-1}^k + \tau_{a_{n-1}^k}), & n=2,\ldots,\eta^k, \\ l^k - (t_{n-1}^k + \tau_{a_{n-1}^k}), & n=\eta^k+1. \end{cases} \tag{2}$$

Associated with a CTSNDP-HC solution $\mathcal{W}$ is a set of consolidation plans, where each consolidation plan defines how a subset of commodities are transported together through an arc of the solution. More precisely, we denote with $\mathcal{C}=\{\mathcal{C}_1,\mathcal{C}_2,\ldots,\mathcal{C}_{|\mathcal{C}|}\}$ a set of consolidation plans. Each $\mathcal{C}_r=(\alpha_r,J_r)$, $r=1,2,\ldots,|\mathcal{C}|$, denotes a consolidation plan for arc $\alpha_r\in\mathcal{A}$, with J_r being a set of pairs (k,n) with $a_n^k=\alpha_r$, indicating that such commodities k are shipped together through arc α_r when they are routed through the n-th arcs of their paths P^k in solution $\mathcal{W}$.

For each arc $\alpha=(i,j)\in\cup_{k\in\mathcal{K}}P^k$, we define $\Theta(\alpha)=\{t_n^k:(k,n)\in J_r,\exists\, r\in\{1,2,\ldots,|\mathcal{C}|\}\text{ s.t. }\alpha_r=\alpha\}$ as the set of departure times associated with the arcs of paths in solution $\mathcal{W}$. Accordingly, for each of such departure times $t\in\Theta(\alpha)$, a consolidation plan $(\alpha,I(\alpha,t))$ can be defined for arc α, where $I(\alpha,t)$ defined below indicates the set of commodities that are shipped together through α with departure time t in solution $\mathcal{W}$:

$$I(\alpha,t)=\{(k,n):k\in\mathcal{K},a_n^k=\alpha\in P^k\text{ and }t_n^k=t\}.$$

With this, a set of consolidation plans $\mathcal{C}$ can be defined by solution $\mathcal{W}$ as follows:

$$\mathcal{C}=\{(\alpha,I(\alpha,t)):\forall\alpha\in\cup_{k\in\mathcal{K}}P^k,t\in\Theta(\alpha),I(\alpha,t)\neq\emptyset\}.$$

The cost $z(\mathcal{W})$ of a CTSNDP-HC solution $\mathcal{W}$ can then be computed as a function of the holding and consolidation plans:

$$z(\mathcal{W})=\sum_{\mathcal{C}_r\in\mathcal{C}} f_{\alpha_r}\left\lceil\frac{\sum_{(k,n)\in J_r}q^k}{u_{\alpha_r}}\right\rceil+\sum_{k\in\mathcal{K}}\sum_{n=1}^{\eta^k}c_{a_n^k}^k q^k+\sum_{k\in\mathcal{K}}\sum_{n=1}^{\eta^k+1}(h_{v_n^k}^k q^k)\delta_n^k, \tag{3}$$

where the three terms represent the fixed, flow and holding costs, respectively.

Alternatively, a feasible CTSNDP-HC solution can also be defined by (i) a routing plan $\mathcal{P}=\{P^k\}_{k\in\mathcal{K}}$, (ii) a set of consolidation plans $\mathcal{C}$ and (iii) a set of departure times $\{t^k\}_{k\in\mathcal{K}}$ which form the consolidation plans $\mathcal{C}$, and such that for each $k\in\mathcal{K}$, path P^k is k-feasible and the waiting times δ^k computed by expressions (2) are non-negative. We define $\mathcal{S}=(\mathcal{P},\mathcal{C})$ as a *flat solution*, and the flat solution $\mathcal{S}$ is *implementable* if a set of departure times $\{t^k\}_{k\in\mathcal{K}}$ satisfying condition (iii) exists.

As an example, Table 1 gives the set of consolidation plans $\mathcal{C}$ associated with the example of Figure 1-(c). For each of the three commodities, the table gives the corresponding path P^k. The set of consolidation plans $\mathcal{C}$ shows three consolidations plans associated with arcs (b,a), (d,b), and (c,b). In particular, all three commodities are consolidated on arc $\alpha=(b,a)$.

Table 1 **Set of consolidation plans for the example of Figure 1-(c)**

k	P^k	$\mathcal{C}$	
		α	J
1	(b,a)	(b,a)	$\{(1,1),(2,2),(3,2)\}$
2	(d,b,a)	(d,b)	$\{(2,1)\}$
3	(c,b,a)	(c,b)	$\{(3,1)\}$

3.2. A time-indexed formulation for the CTSNDP-HC

Like many other flow-over-time problems (see, for example, Fleischer and Skutella (2007) and Skutella (2009)) and network design problems (see, for example, Andersen et al. (2009*b*), Andersen et al. (2009*a*), and Pedersen et al. (2009)), the CTSNDP-HC can be approximated by a time-expanded network formulation.

We consider a time-expanded network with a discretization level Δ, $\mathcal{D}^{\Delta}_{\mathcal{T}} = (\mathcal{N}^{\Delta}_{\mathcal{T}}, \mathcal{H}^{\Delta}_{\mathcal{T}} \cup \mathcal{A}^{\Delta}_{\mathcal{T}})$ where $\mathcal{T} = (\mathcal{T}_i)_{i\in\mathcal{N}}$ is a set of time points with $\mathcal{T}_i = \{0, \Delta, 2\Delta, ..., M\Delta\}$ for all $i \in \mathcal{N}$ and for $M \in \mathbb{N}_{>0}$ with $M = max_{k\in\mathcal{K}}\{\lceil l^k/\Delta \rceil\}$. The node set $\mathcal{N}^{\Delta}_{\mathcal{T}}$ has a node (i,t) for each $i \in \mathcal{N}$ and $t \in \mathcal{T}_i$. The set of arcs of $\mathcal{D}^{\Delta}_{\mathcal{T}}$ contains two subsets of arcs:

- *Holding* arcs $\mathcal{H}^{\Delta}_{\mathcal{T}}$. For every node $i \in \mathcal{N}$, and every $t \in \mathcal{T}_i \setminus \{M\Delta\}$, there is an arc $((i,t),(i,t+\Delta))$ representing a holding time of Δ time units at node i.
- *Dispatch* or *service* arcs $\mathcal{A}^{\Delta}_{\mathcal{T}}$. For every arc $(i,j) \in \mathcal{A}$, and every node $(i,t) \in \mathcal{N}^{\Delta}_{\mathcal{T}}$, there is an arc $((i,t),(j,\bar{t}))$ with $i \neq j$ representing a dispatch from node i at time t arriving at time $\bar{t}$ at node j with $\bar{t} = t + \Delta\lceil \tau_{ij}/\Delta \rceil$ and $\bar{t} \leq M\Delta$, and since $\tau_{ij} \leq \Delta\lceil \tau_{ij}/\Delta \rceil$, the condition $\bar{t} \geq t + \tau_{ij}$ holds, thus guaranteeing that the feasible solutions of a TI formulation based on graph $\mathcal{D}^{\Delta}_{\mathcal{T}}$ (see below) are also feasible for the CTSNDP-HC.

Network $\mathcal{D}^{\Delta}_{\mathcal{T}}$ is also known in the literature as a *condensed* time-expanded network. Below, we model the CTSNDP-HC using a TI formulation based on graph $\mathcal{D}^{\Delta}_{\mathcal{T}}$, denoted as SND-HC($\mathcal{D}^{\Delta}_{\mathcal{T}}$).

Let $y^{t\bar{t}}_{ij}$ be a nonnegative integer variable representing the number of times that arc $(i,j) \in \mathcal{A}$ is used to serve the dispatches from node i at time t arriving at time $\bar{t}$ in j, and let $x^{kt\bar{t}}_{ij}$ be 0-1 variable equal to 1 if commodity $k \in \mathcal{K}$ is routed along arc $(i,j) \in \mathcal{A}$ departing from i at time t and arriving at j at time $\bar{t}$, 0 otherwise. Moreover, let w^k_i be a nonnegative variable denoting the holding or waiting time of commodity k at node i. Formulation SND-HC($\mathcal{D}^{\Delta}_{\mathcal{T}}$) is as follows:

$$z(\mathcal{D}^{\Delta}_{\mathcal{T}}) = \min \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}^{\Delta}_{\mathcal{T}}} f_{ij} y^{t\bar{t}}_{ij} + \sum_{k\in\mathcal{K}} \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}^{\Delta}_{\mathcal{T}}} (c^k_{ij} q^k) x^{kt\bar{t}}_{ij} + \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}} (h^k_i q^k) w^k_i \tag{4}$$

$$s.t. \quad \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}^{\Delta}_{\mathcal{T}}\cup\mathcal{H}^{\Delta}_{\mathcal{T}}} x^{kt\bar{t}}_{ij} - \sum_{((j,\bar{t}),(i,t))\in\mathcal{A}^{\Delta}_{\mathcal{T}}\cup\mathcal{H}^{\Delta}_{\mathcal{T}}} x^{k\bar{t}t}_{ji} = \begin{cases} 1 & (i,t) = (o^k, e^k), \\ -1 & (i,t) = (d^k, l^k), \\ 0 & \text{otherwise}, \end{cases} \quad \forall k \in \mathcal{K}, (i,t) \in \mathcal{N}^{\Delta}_{\mathcal{T}}, \tag{5}$$

$$\sum_{k\in\mathcal{K}} q^k x^{kt\bar{t}}_{ij} \leq u_{ij} y^{t\bar{t}}_{ij}, \quad \forall\ ((i,t),(j,\bar{t})) \in \mathcal{A}^{\Delta}_{\mathcal{T}}, \tag{6}$$

$$w_i^k = \begin{cases} \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}^{\Delta}} t\, x_{ij}^{kt\bar{t}} - e^k, & i = o^k, \\ l^k - \sum_{((j,\bar{t}),(i,t))\in\mathcal{A}_{\mathcal{T}}^{\Delta}} t\, x_{ji}^{k\bar{t}t}, & i = d^k, \\ \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}^{\Delta}} t\, x_{ij}^{kt\bar{t}} - \sum_{((j,\bar{t}),(i,t))\in\mathcal{A}_{\mathcal{T}}^{\Delta}} t\, x_{ji}^{k\bar{t}t}, & \text{otherwise}, \end{cases} \quad \forall\, i\in\mathcal{N}, \forall\, k\in\mathcal{K}, \tag{7}$$

$$x_{ij}^{kt\bar{t}} \in \{0,1\}, \quad \forall((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}^{\Delta}\cup\mathcal{H}_{\mathcal{T}}^{\Delta}, k\in\mathcal{K}, \tag{8}$$

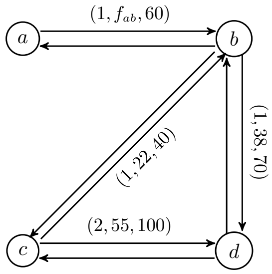
$$y_{ij}^{t\bar{t}} \in \mathbb{N}_{\geq 0}, \quad \forall((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}^{\Delta}, \tag{9}$$

$$w_i^k \geq 0, \quad \forall i\in\mathcal{N}, k\in\mathcal{K}. \tag{10}$$

For notation convenience, we assume that the node set $\mathcal{N}_{\mathcal{T}}^{\Delta}$ also contains nodes (o^k, e^k) and (d^k, l^k) for all $k\in\mathcal{K}$. Otherwise, the nodes $(o_k, \Delta\lceil e^k/\Delta\rceil)$ and $(d_k, \Delta\lfloor l^k/\Delta\rfloor)$ can be used instead in Constraints (5). In the above formulation, the objective function (4) aims to minimize the total cost computed as the sum of the fixed, flow and holding costs, respectively. Constraints (5) are *flow conservation constraints* ensuring that each commodity $k\in\mathcal{K}$ is routed along a single path starting from its origin and ending at its destination before its due time. Each commodity $k\in\mathcal{K}$ departs from o^k at time e^k and arrives at d^k at time l^k, and holding arcs allow a commodity to arrive early at its destination or depart later from its origin. Constraints (6) ensure that the flow on each service arc does not exceed the capacity installed on the arc. Constraints (7) define the values of variables w_i^k, computed as the difference between the departure and arrival times. Note that in each summation of the first two expressions involving variables $x_{ij}^{kt\bar{t}}$, exactly one variable x is equal to 1 in any optimal solution, therefore the summation produces the time t defined by the variable. Regarding the third expression, because we do not rule out non-elementary paths in the solution to cover more general case, the expression computes the difference between the sum of the departure times and the sum of the arrival times. Finally, constraints (8), (9) and (10) state the domains of the decision variables. The above formulation contains as a special case the TI formulation described by Boland et al. (2017) and used to solve the CTSNDP. Indeed, when each coefficient h_i^k is assumed to be equal to zero, the corresponding decision variables w_i^k are not necessary in the formulation. We denote with SND($\mathcal{D}_{\mathcal{T}}^{\Delta}$) the resulting formulation.

Due to the definition of the time-expanded network $\mathcal{D}_{\mathcal{T}}^{\Delta}$ such that for each arc $(i,j)\in\mathcal{A}$ we have $\Delta\lceil\tau_{ij}/\Delta\rceil \geq \tau_{ij}$, and due to the requirement that for every commodity k, its path starts at time e^k and ends at time l^k, any feasible solution of formulation SND-HC($\mathcal{D}_{\mathcal{T}}^{\Delta}$) is also a feasible solution for the CTSNDP-HC. Nevertheless, an optimal SND-HC($\mathcal{D}_{\mathcal{T}}^{\Delta}$) solution is not necessarily an optimal CTSNDP-HC solution, due to the discretization factor Δ. Moreover, SND-HC($\mathcal{D}_{\mathcal{T}}^{\Delta}$) may be infeasible even if the original CTSNDP-HC is feasible.

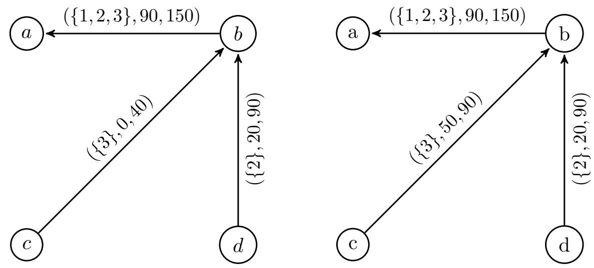
It is not straightforward to prove the existence of a value $\hat{\Delta}$ such that $z(\mathcal{D}_{\mathcal{T}}^{\hat{\Delta}})$ provides the optimal solution cost of the CTSNDP-HC. Indeed, as observed by Boland and Savelsbergh (2019), for some problems such as the TSPTW, a simple combinatorial argument suffices to show the existence of



(a) Network $\mathcal{D}$: (c, f, τ) on arcs

k	o^k	d^k	q^k	e^k	l^k
1	b	a	25	0	160
2	d	a	30	20	180
3	c	a	40	0	180

(b) Instance data



(c) An optimal solution for the CTSNDP

(d) Alternative optimal solution for the CTSNDP

Figure 2 **Examples showing that for the CTSNDP-HC the transformation of Boland et al. (2017) cannot be applied (Figures (c) and (d):** $(\{commodities\}, dep.time, arr.time)$ **on arcs)**

a complete TI model, whereas for other problems, such as the continuous-time inventory routing problem (Lagos et al. 2020), such $\hat{\Delta}$ may be smaller than one and is difficult to identify.

3.3. Existence of a finite time-expanded network for the CTSNDP-HC

For the CTSNDP, the existence of a complete TI model has been shown by Boland et al. (2017), who noted that when the travel times and time window limits are integer-valued, the set of times points e^k, for some commodity $k \in \mathcal{K}$, or of the form $e^k + \sum_{a\in P} \tau_a$, for some path P originating at o^k, suffice to compose a complete TI model. The observation is that the dispatch times of a path P can be shifted to be as early as possible without changing any consolidations so that the total cost is not changed, and strictly relies on the assumption that in-storage holding costs are equal to zero. For the CTSNDP-HC, due to the presence of nonzero holding costs in the problem objective, such an observation is no longer valid.

To illustrate the case, Figure 2 considers the example of Figure 1 where the solution of Figure 2-(c) depicts an optimal solution of the CTSNDP. The alternative CTSNDP optimal solution represented by Figure 2-(d), where the departure time at the origin of commodity 3 is equal to 50, can be shifted to be as early as possible, thus obtaining the departure time of the solution of Figure 2-(c) without changing the consolidations on arc (a, b). When considering the CTSNDP-HC, if the per-unit-of-demand-and-time holding costs for commodity 3 at terminals b and c are equal to 0.01 and 0.005, respectively, commodity 3 incurs a holding cost equal to $0.01 \times 50 \times 40$ for the solution of Figure 2-(c), whereas it incurs a lower holding cost equal to $0.005 \times 50 \times 40$ for the solution of Figure 2-(d). Therefore, the time point $(c, 50)$, which is not part of a complete TI model for the CTSNDP, must be considered when solving the CTSNDP-HC.

In order to show that a complete TI model exists for the CTSNDP-HC, it is necessary to prove that, given a flat solution $\mathcal{S}$, a linear program (LP) can be defined to determine optimal departure times t^k for each $k \in \mathcal{K}$ that are integers. This argument suffices to show that a complete TI model exists. The LP argument was also used by Boland et al. (2015) for a network scheduling problem.

Consider a flat solution $\mathcal{S} = (\mathcal{P}, \mathcal{C})$, with a routing plan $\mathcal{P}$ and a set of consolidation plans $\mathcal{C}$ associated with $\mathcal{P}$. We denote with $z_{fc}(\mathcal{S})$ the cost of the flat solution $\mathcal{S}$, that is, the sum of its fixed and flow costs:

$$z_{fc}(\mathcal{S}) = \sum_{\mathcal{C}_r \in \mathcal{C}} f_{\alpha_r} \left\lceil \frac{\sum_{(k,n)\in J_r} q^k}{u_{\alpha_r}} \right\rceil + \sum_{k\in\mathcal{K}} \sum_{n=1}^{\eta^k} c^k_{a^k_n} q^k. \tag{11}$$

For each $k \in \mathcal{K}$ and each node $v^k_n \in P^k$, $n = 1, 2, ..., \eta^k + 1$, we define nonnegative continuous variables $\pi^k_{v_n}$ and $t^k_{v_n}$ as the arrival and departure times of commodity k at node v^k_n, respectively. Moreover, for each $\mathcal{C}_r = (\alpha_r, J_r)$, $r = 1, ..., |\mathcal{C}|$, we define a nonnegative continuous variable $\hat{t}_{\mathcal{C}_r} \geq 0$ representing the consolidation time of the commodities in J_r on arc α_r, i.e., the joint departure time of all commodities $(k, n) \in J_r$.

If the flat solution $\mathcal{S}$ is implementable, then the following LP formulation, denoted as *implementable model* ($IM(\mathcal{S})$), computes corresponding optimal departure times for flat solution $\mathcal{S}$ of cost $z_{fc}(\mathcal{S}) + z_w(\mathcal{S})$:

$$z_w(\mathcal{S}) = \min \sum_{k\in\mathcal{K}} \sum_{n=1}^{\eta^k+1} h^k_{v_n} q^k (t^k_{v_n} - \pi^k_{v_n}) \tag{12a}$$

$$s.t. \quad \pi^k_{v_{n+1}} - t^k_{v_n} = \tau_{v_n v_{n+1}}, \quad \forall\, k \in \mathcal{K},\, n = 1, ..., \eta^k, \tag{12b}$$

$$t^k_{v_n} - \pi^k_{v_n} \geq 0, \quad \forall k \in \mathcal{K},\, n = 1, ..., \eta^k + 1, \tag{12c}$$

$$\hat{t}_{\mathcal{C}_r} - t^k_{v_n} = 0, \quad \forall (k, n) \in J_r,\, r = 1, ..., |\mathcal{C}|, \tag{12d}$$

$$\pi^k_{o^k} = e^k, \quad \forall k \in \mathcal{K}, \tag{12e}$$

$$t^k_{d^k} = l^k, \quad \forall k \in \mathcal{K}, \tag{12f}$$

$$\pi^k_{v_n} \geq 0, \quad \forall\, k \in \mathcal{K},\, n = 1, ..., \eta^k + 1, \tag{12g}$$

$$t^k_{v_n} \geq 0, \quad \forall k \in \mathcal{K},\, n = 1, ..., \eta^k + 1, \tag{12h}$$

$$\hat{t}_{\mathcal{C}_r} \geq 0, \quad \forall r = 1, ..., |\mathcal{C}|. \tag{12i}$$

The objective function (12a) aims to minimize the total holding cost associated with the flat solution. Constraints (12b), (12c), (12e) and (12f) define the arrival and departure times according to path P^k, respectively. Constraints (12d) impose that the departure times at the intermediate nodes follow the set of consolidation plans $\mathcal{C}$.

The following proposition implies the existence of a complete TI model for the CTSNDP-HC.

Proposition 1 *If e_k, l_k and τ_{ij} are integer-valued, and the flat solution $\mathcal{S} = (\mathcal{P}, \mathcal{C})$ is implementable, then for formulation IM($\mathcal{S}$) there is an integral optimal solution.*

Proof. The proof is provided in §EC.1.1 in the e-companion to this paper. □

Algorithm 1: DDD algorithm for the CTSNDP

Input: CTSNDP defined on a flat network $\mathcal{D} = (\mathcal{N}, \mathcal{A})$
Output: Solution $\mathcal{W} = \{\mathcal{W}^k\}_{k \in \mathcal{K}}$ of cost UB
begin
  // Initialization
  $UB \leftarrow +\infty$, $LB \leftarrow -\infty$, $gap \leftarrow +\infty$, $\mathcal{W} \leftarrow \emptyset$;
  // Initialize the partially time-expanded network
  $\mathcal{D}_\mathcal{T} \leftarrow (\mathcal{N}_\mathcal{T}, \mathcal{H}_\mathcal{T} \cup \mathcal{A}_\mathcal{T})$;
  // Termination condition
  while $gap > optimality\ tolerance$ **do**
    // Solution of the relaxation
    Solve SND($\mathcal{D}_\mathcal{T}$) and set LB equal to the optimal solution cost of SND($\mathcal{D}_\mathcal{T}$);
    // Compute a feasible CTSNDP solution
    Compute a valid upper bound z based on the solution defined by the relaxation;
    if $z < UB$ **then**
      $UB \leftarrow z$;
      Update solution $\mathcal{W}$;
    end
    // Compute the optimality tolerance
    $gap \leftarrow (UB - LB)/UB$;
    // Check the optimality condition
    if $gap > optimality\ tolerance$ **then**
      // Optimality not reached
      Based on the solution of SND($\mathcal{D}_\mathcal{T}$), refine the network $\mathcal{D}_\mathcal{T}$ to correct the length of at least one *short arc*;
    end
  end
  return Solution $\mathcal{W}$ of cost UB;
end

Notice that the above proposition does not suggest a specific value of the discretization Δ. It is easy to see that if ratios τ_{ij}/Δ, e_k/Δ and l_k/Δ are integer-valued for some $\Delta \in \mathbb{N}_{>0}$, $z(\mathcal{D}_\mathcal{T}^\Delta)$ corresponds to the optimal solution cost of the CTSNDP-HC, and that in the worst case we have $\Delta = 1$. In practice, the size of the complete TI model can be computationally intractable, but in the next section we describe an exact algorithm aimed at finding the optimal CTSNDP-HC solution by solving a set of reduced models of the complete TI model.

4. An exact algorithm for the CTSNDP-HC

A complete TI model for the CTSNDP-HC implies the existence of a discretization $\hat{\Delta}$ and of the corresponding fully time-expanded network $\mathcal{D}_\mathcal{T}^{\hat{\Delta}}$ such that the optimal solution cost $z(\mathcal{D}_\mathcal{T}^{\hat{\Delta}})$ of formulation SND-HC($\mathcal{D}_\mathcal{T}^{\hat{\Delta}}$) is the optimal solution cost of the CTSNDP-HC. However, the size of the network $\mathcal{D}_\mathcal{T}^{\hat{\Delta}}$ can be prohibitively large, and the resulting TI model impractical to be solved by conventional techniques.

For the CTSNDP, Boland et al. (2017) proposed a DDD algorithm that dynamically and iteratively determines the time points that are present in an optimal solution. Let $\mathcal{D}_\mathcal{T} = (\mathcal{N}_\mathcal{T}, \mathcal{H}_\mathcal{T} \cup$

$\mathcal{A}_{\mathcal{T}}$) be a partially time-expanded network such that $|\mathcal{N}_{\mathcal{T}}| \ll |\mathcal{N}_{\hat{\mathcal{T}}}^{\hat{\Delta}}|$. As illustrated in Algorithm 1, the DDD algorithm starts by properly initializing the partial network $\mathcal{D}_{\mathcal{T}}$, and at each iteration of the algorithm, a relaxation model SND($\mathcal{D}_{\mathcal{T}}$) is solved to compute a valid lower bound LB on the CTSNDP. An upper bound is also computed using the lower bound solution and the algorithm iterates until a predefined optimality tolerance is reached. The partially time-expanded network $\mathcal{D}_{\mathcal{T}}$ is initialed and modified (or *refined*) whenever the optimality tolerance is not reached to ensure the computation of a valid lower bound. The construction of the network $\mathcal{D}_{\mathcal{T}}$, together with the relaxation model SND($\mathcal{D}_{\mathcal{T}}$), ensures that whenever the lower bound solution is not proved to be an optimal CTSNDP solution, the network contains at least one arc, say arc $((i,t),(j,t'))$ such that $t' < t + \tau_{ij}$, i.e., the length of the arc is *too short* and arc $((i,t),(j,t'))$ is a *short-arc*. The network is therefore refined by correcting the length of arc $((i,t),(j,t'))$, adding new time points and corresponding arcs. The correctness of the algorithm follows on from the validity of bounds LB and UB. The convergence of the method relies on the refinement strategies, which guarantee that the final relaxation model will eventually converge to the complete TI model.

In this paper, we adapt the DDD Algorithm 1 in a novel way to solve the more general CTSNDP-HC as summarized below.

(i) At Step 4, we solve a novel relaxation of formulation SND-HC($\mathcal{D}_{\hat{\mathcal{T}}}^{\hat{\Delta}}$). The relaxation relies on both the definition of the network $\mathcal{D}_{\mathcal{T}}$ and on a formulation obtained by relaxing equations (7) defining the holding variables w_i^k (see §4.1).

(ii) At Step 5, we compute a valid upper bound UB based on the flat solution $\mathcal{S}$ defined by the relaxation using a new heuristic algorithm accounting for the holding costs (see §4.3).

(iii) At Step 12, we extend the refinement strategy used by Boland et al. (2017) to add new time points based on the definition of variables w_i^k (see §4.4).

In the reminder of this section, we give the details of above three main components (i)–(iii) of the DDD algorithm adapted for the CTSNDP-HC. These three components play a crucial role in the effectiveness of the DDD algorithm for the CTSNDP-HC, and require substantial changes to the original components designed for the CTSNDP. We conclude the section by proving that the exact DDD algorithm solves the CTSNDP-HC to optimality in a finite number of iterations (§4.5).

In what follow, we assume that all time data are integer-valued, and based on Proposition 1, $\hat{\Delta}$ assumes a value equal to 1.

4.1. A relaxation of the CTSNDP-HC

In this section, we describe a valid relaxation for the CTSNDP-HC. We start by a simple observation that if $h_i^k = 0$, $\forall k \in \mathcal{K}, i \in \mathcal{N}$, then the CTSNDP-HC reduces to the CTSNDP, and any valid lower bound (including the optimal objective value) for the CTSNDP is also a valid lower bound for the

CTSNDP-HC. Boland et al. (2017) described a lower bound for the CTSNDP based on a special case of formulation SND-HC($\mathcal{D}_\mathcal{T}$) where variables w_i^k and constraints (7) are disregarded, since in-storage holding costs h_i^k are all equal to zero. They show that the following three properties of the partial network $\mathcal{D}_\mathcal{T}$ suffice to provide a valid lower bound on $z(\mathcal{D}_{\hat{\mathcal{T}}})$.

Property 1 *For all commodities $k \in \mathcal{K}$, the nodes (o^k, e^k) and (d^k, l^k) are in $\mathcal{N}_\mathcal{T}$.*

Property 2 *Every arc $((i,t),(j,\bar{t})) \in \mathcal{A}_\mathcal{T}$ has $\bar{t} \le t + \tau_{ij}$.*

Property 3 *For every arc $a = (i,j) \in \mathcal{A}$ in the flat network $\mathcal{D}$, and for every node (i,t) in the partial network $\mathcal{D}_\mathcal{T}$, there is an arc $a' = ((i,t),(j,\bar{t}))$ in $\mathcal{D}_\mathcal{T}$ for some $\bar{t} \in \mathcal{N}_\mathcal{T}$ (a' is defined as a timed copy of arc a in $\mathcal{A}_\mathcal{T}$).*

Let $(\bar{x}, \bar{y})$ be an optimal solution of formulation SND-HC($\mathcal{D}_{\hat{\mathcal{T}}}$) with zero holding costs (i.e., an optimal CTSNDP solution) and let $\overline{\mathcal{A}} = \{((i,t),(j,t+\tau_{ij})) \in \mathcal{A}_{\hat{\mathcal{T}}} : \bar{y}_{ij}^{t,t+\tau_{ij}} > 0\}$ be the set of arcs traversed by the commodities in the solution. For any arc $a = ((i,t),(j,t+\tau_{ij})) \in \overline{\mathcal{A}}$, define $\rho_i(t) = \arg\max\{s \in \mathcal{T}_i : s \le t\}$. The existence of $\rho_i(t)$ is ensured by the three properties. Indeed, denote $\underline{\tau}_{ij}$ as being the length of any shortest-time path from i to j in the flat network $\mathcal{D}$. Then, for each $k \in \mathcal{K}$, and each $i \in \mathcal{N}$ a node $(i,t) \in \mathcal{N}_\mathcal{T}$ exists with $t \le e_k + \underline{\tau}_{o_k i}$. Further, by Property 3 a timed-copy arc $((i,\rho_i(t)),(j,t')) \in \mathcal{A}_\mathcal{T}$ of arc $a = ((i,t),(j,t+\tau_{ij})) \in \overline{\mathcal{A}}$ exists in $\mathcal{D}_\mathcal{T}$ for some $(j,t') \in \mathcal{N}_\mathcal{T}$, and define $\sigma(a)$ to be any such t'. Based on these, Proposition 2 below shows that formulation SND-HC($\mathcal{D}_\mathcal{T}$) with zero holding costs defined over a network $\mathcal{D}_\mathcal{T}$ satisfying Properties 1, 2 and 3 is a valid relaxation for formulation SND-HC($\mathcal{D}_{\hat{\mathcal{T}}}$).

Proposition 2 (Boland et al. (2017)) *If the in-storage holding costs h_i^k are all equal to zero, the mapping of solution $(\bar{x},\bar{y})$ into a solution (x,y) of formulation SND-HC($\mathcal{D}_\mathcal{T}$) defined by $\mu : \overline{\mathcal{A}} \to \mathcal{A}_\mathcal{T}$ with $\mu(a) = ((i,\rho_i(t)),(j,\sigma(a)))$ and computed by the following expressions for each $\tilde{a} = ((i,\tilde{t}),(j,\tilde{t}')) \in \mathcal{A}_\mathcal{T}$:*

$$x_{ij}^{k\tilde{t}\tilde{t}'} = \sum_{\substack{a=((i,t),(j,t+\tau_{ij}))\in\overline{\mathcal{A}}:\\ \mu(a)=\tilde{a}}} \bar{x}_{ij}^{kt,t+\tau_{ij}} \quad \text{and} \quad y_{ij}^{\tilde{t}\tilde{t}'} = \sum_{\substack{a=((i,t),(j,t+\tau_{ij}))\in\overline{\mathcal{A}}:\\ \mu(a)=\tilde{a}}} \bar{y}_{ij}^{t,t+\tau_{ij}}, \tag{13}$$

corresponds to a feasible solution of formulation SND-HC($\mathcal{D}_\mathcal{T}$) of the same cost of solution $(\bar{x},\bar{y})$.

The proof of the above proposition is based on the observation that, for each commodity $k \in \mathcal{K}$, to each path $\overline{P}^k = (a_1^k, \ldots, a_{\eta^k}^k)$, $a_h^k \in \overline{\mathcal{A}}$, $h = 1, \ldots, \eta^k$, induced by solution $(\bar{x},\bar{y})$ with $a_h^k = ((i_h^k, t_h^k),(i_{h+1}^k, t_h^k + \tau_{i_h^k i_{h+1}^k}))$ and $t_{h+1}^k \ge t_h^k + \tau_{i_h^k i_{h+1}^k}$ for $h = 1,..,\eta^k - 1$, there is a corresponding feasible path $P^k = (\mu(a_1^k), \ldots, \mu(a_{\eta^k}^k))$ in $\mathcal{D}_\mathcal{T}$ with appropriate holding arcs. Figure 3 illustrates the

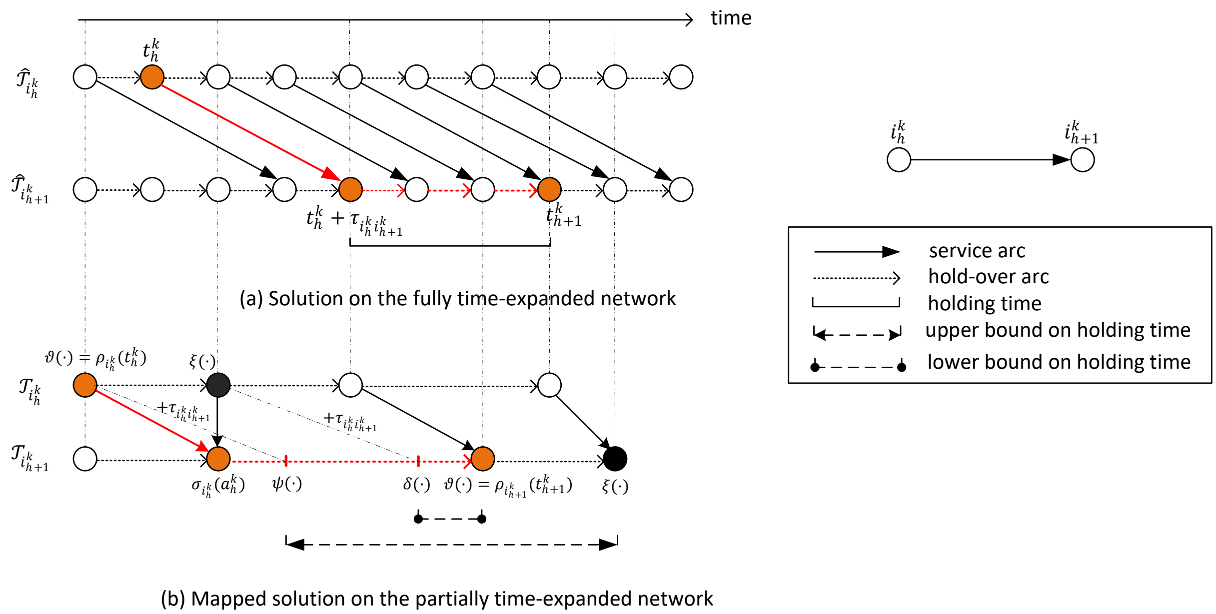


Figure 3 **Mapping functions and bounds on holding time**

mapping for arc a_h^k. The above proposition relies on the fact that adding additional holding arcs does not result in additional costs. As Figure 3 shows, the holding time $w^k_{i^k_{h+1}} = \rho_{i^k_{h+1}}(t^k_{h+1}) - \sigma_{i^k_h}(a^k_h)$ of the partially time-expanded network can be larger than the true holding time $\overline{w}^k_{i^k_{h+1}} = t^k_{h+1} - (t^k_h + \tau_{i^k_h i^k_{h+1}})$ of the fully time-expanded network. Therefore, in the presence of a nonzero holding cost, the holding cost associated with solution (x, y, w) is not always less than or equal to the holding cost associated with solution $(\bar{x}, \bar{y}, \bar{w})$. Thus, even under Properties 1-3, formulation SND-HC($\mathcal{D}_{\mathcal{T}}$) does not provide a valid relaxation for formulation SND-HC($\mathcal{D}_{\hat{\mathcal{T}}}$).

To obtain a valid relaxation for formulation SND-HC($\mathcal{D}_{\hat{\mathcal{T}}}$), we derive a relaxation of equations (7) based on the following observations. Let $P^k = (a^k_1, \ldots, a^k_{\eta^k})$ with $a^k_h = ((i^k_h, t^k_h), (i^k_{h+1}, \bar{t}^k_{h+1}))$, $h = 1, \ldots, \eta^k$, $k \in \mathcal{K}$, be a path in network $\mathcal{D}_{\hat{\mathcal{T}}}$ representing a feasible k-path, where we denote with $\bar{t}^k_h$ and t^k_h the arrival and departure times at node i^k_h, $h = 1, .., \eta^k + 1$, respectively:

(i) On the partial network $\mathcal{D}_{\mathcal{T}}$ satisfying Properties 1-3, with each arrival time $\bar{t}^k_h$, $h = 2, .., \eta^k + 1$, we can associate a lower bound $\check{t}_{arr} \le \bar{t}^k_h$ computed as $\check{t}_{arr} = \psi^k(\mu(a^k_{h-1}))$ and an upper bound $\hat{t}_{arr} \ge \bar{t}^k_h$ computed as $\hat{t}_{arr} = \delta^k(\mu(a^k_{h-1}))$. In addition, with each departure time t^k_h, we can associate an upper bound $\hat{t}_{dep} \ge t^k_h$ computed as $\hat{t}_{dep} = \xi^k(\mu(a^k_h))$ and a lower bound $\check{t}_{dep} \le t^k_h$ computed as $\check{t}_{dep} = \vartheta^k(\mu(a^k_h))$. With these, we can obtain an upper bound on the holding time at node i_h, i.e., $t^k_h - \bar{t}^k_h \le \hat{t}_{arr} - \check{t}_{dep}$, as well as a lower bound on the holding time at node i_h, i.e., $t^k_h - \bar{t}^k_h \ge \check{t}_{arr} - \hat{t}_{dep}$.

(ii) Let $T(P^k)$ be the total transit time of path P^k, computed as $T(P^k) = \sum_{h=1}^{\eta^k} \tau_{a^k_h}$. Then, the total holding time of path P^k must be equal to $l^k - e^k - T(P^k)$ since each commodity $k \in \mathcal{K}$ leaves its origin o^k at time e^k and arrives at its destination d^k at time l^k.

For a commodity $k \in \mathcal{K}$, let $\phi^k(i,j)$ denote the time of the shortest-time path from node i to node j in the flat network $\mathcal{D}$ with the reduced arc set $\mathcal{A} \setminus (\{(i,o^k) \in \mathcal{A} : i \in \mathcal{N}\} \cup \{(d^k,j) \in \mathcal{A} : j \in \mathcal{N}\})$ computed with respect to travel times τ_{ij}.

Since the true arrival and departure times must be larger than the arrival time along the shortest-path, the two lower bound functions $\psi^k(a)$ and $\vartheta^k(a)$ for each $k \in \mathcal{K}$ and $a = ((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}$ can be computed as:

$$\psi^k(a) = \max\{t + \tau_{ij}, e^k + \phi^k(o^k,i) + \tau_{ij}\}, \tag{14}$$

$$\vartheta^k(a) = \max\{t, e^k + \phi^k(o^k,i)\}. \tag{15}$$

Further, for $i \in \mathcal{N}$, let $\vec{t}_i(t)$ be the next time point of point t in set $\mathcal{T}_i$ computed as

$$\vec{t}_i(t) = \begin{cases} \arg\min\{t' \in \mathcal{T}_i : t' > t\}, \text{ if } t < \arg\max\{t' \in \mathcal{T}_i\}, \\ \max_{k \in \mathcal{K}}(l^k - \phi^k(i,d^k)), \text{ if } t = \arg\max\{t' \in \mathcal{T}_i\}, \end{cases} \tag{16}$$

where the term $l^k - \phi^k(i,d^k)$ represents the latest departure time from node i for commodity k to arrive at its destination d^k before l^k. Function $\xi^k(a)$ for for each $k \in \mathcal{K}$ and $a = ((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}$, can be computed as:

$$\xi^k(a) = \begin{cases} \min\{\vec{t}_i(t), l^k - \tau_{ij} - \phi^k(j,d^k)\}, & \text{if } \vec{t}_i(t) - t > 1, \\ \min\{t, l^k - \tau_{ij} - \phi^k(j,d^k)\}, & \text{otherwise.} \end{cases} \tag{17}$$

Based on this, function $\delta^k(a)$ can be computed as:

$$\delta^k(a) = \xi^k(a) + \tau_{ij}. \tag{18}$$

If the commodity k passes through arc $a = ((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}$, functions $\psi^k(a)$ and $\delta^k(a)$ compute the lower and upper bounds of the true arrival time at terminal j, respectively. In contrast, functions $\vartheta^k(a)$ and $\xi^k(a)$ calculate the lower and upper bounds of the true departure time from terminal i, respectively. Figure 3 gives an example of how these four functions produce valid lower and upper bounds for the true holding time.

Based on the above observations, we obtain the following relaxation of formulation SND-HC($\mathcal{D}_{\hat{\mathcal{T}}}$), called SND-HC-R($\mathcal{D}_{\mathcal{T}}$):

$$z_R(\mathcal{D}_{\mathcal{T}}) = \min \sum_{((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}} f_{ij} y_{ij}^{t\bar{t}} + \sum_{k \in \mathcal{K}} \sum_{((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}} (c_{ij}^k q^k) x_{ij}^{kt\bar{t}} + \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{N}} (h_i^k q^k) w_i^k \tag{19}$$

$$s.t. \quad (5), (6), (8), (9), (10) \text{ and} \tag{20}$$

$$w_i^k \le \begin{cases} \sum_{a=((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}} \xi^k(a) x_{ij}^{kt\bar{t}} - e^k, & i = o^k, \\ l^k - \sum_{a=((j,\bar{t}),(i,t)) \in \mathcal{A}_{\mathcal{T}}} \psi^k(a)\, x_{ji}^{k\bar{t}t}, & i = d^k, \\ \sum_{a=((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}} \xi^k(a) x_{ij}^{kt\bar{t}} - \sum_{a=((j,\bar{t}),(i,t)) \in \mathcal{A}_{\mathcal{T}}} \psi^k(a)\, x_{ji}^{k\bar{t}t}, & \text{otherwise,} \end{cases} \quad \forall\, i \in \mathcal{N}, \forall\, k \in \mathcal{K}, \tag{21}$$

$$w_i^k \ge \begin{cases} \sum_{a=((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}} \vartheta^k(a) x_{ij}^{kt\bar{t}} - e^k, & i = o^k, \\ l^k - \sum_{a=((j,\bar{t}),(i,t)) \in \mathcal{A}_{\mathcal{T}}} \delta^k(a)\, x_{ji}^{k\bar{t}t}, & i = d^k, \\ \sum_{a=((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}} \vartheta^k(a) x_{ij}^{kt\bar{t}} - \sum_{a=((j,\bar{t}),(i,t)) \in \mathcal{A}_{\mathcal{T}}} \delta^k(a)\, x_{ji}^{k\bar{t}t}, & \text{otherwise,} \end{cases} \quad \forall\, i \in \mathcal{N}, \forall\, k \in \mathcal{K}, \tag{22}$$

$$\sum_{i \in \mathcal{N}} w_i^k = l^k - e^k - \sum_{((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}} \tau_{ij} x_{ij}^{kt\bar{t}}, \quad \forall\, k \in \mathcal{K}. \tag{23}$$

Inequalities (21) and (22) relax equations (7), whereas constraints (23) impose the total holding time of each commodity $k \in \mathcal{K}$. The following Theorem 1 holds. The lower bound LB of the exact algorithm is thus computed as $LB = z_R(\mathcal{D}_\mathcal{T})$ where, in the computational results reported in §5, relaxation SND-HC-R($\mathcal{D}_\mathcal{T}$) is solved to optimality by means of a general MIP solver.

Theorem 1 *If the partial time-expanded network $\mathcal{D}_\mathcal{T}$ satisfies Properties 1-3, then $z_R(\mathcal{D}_\mathcal{T}) \le z(\mathcal{D}_{\hat{\mathcal{T}}})$, i.e., SND-HC-R($\mathcal{D}_\mathcal{T}$) is a valid relaxation of SND-HC($\mathcal{D}_{\hat{\mathcal{T}}}$).*

Proof. The proof is provided in §EC.1.2 in the e-companion to this paper. □

The quality of relaxation SND-HC-R($\mathcal{D}_\mathcal{T}$) strongly affects the effectiveness of the DDD algorithm for solving the CTSNDP-HC. Here, we further describe ways to strengthen relaxation SND-HC-R($\mathcal{D}_\mathcal{T}$) in order to obtain tighter lower bounds.

As discussed by Boland et al. (2017), the CTSNDP relaxation can be strengthened by means of the following additional property, based on which we can establish Theorem 2 below.

Property 4 (Longest-feasible-arc) *Network $\mathcal{D}_\mathcal{T}$ satisfies the longest-feasible-arc property if for each arc $((i,t),(j,t')) \in \mathcal{A}_\mathcal{T}$ there does not exist a node $(j,t'') \in \mathcal{N}_\mathcal{T}$ with $t' < t'' \le t + \tau_{ij}$, i.e., $t' = \arg max\{s : s \le t + \tau_{ij}, (j,s) \in \mathcal{N}_\mathcal{T}\}$.*

Theorem 2 *For a fixed set of time points $\mathcal{N}_\mathcal{T}$, among the partial time-expanded networks $\mathcal{D}_\mathcal{T} = (\mathcal{N}_\mathcal{T}, \mathcal{H}_\mathcal{T} \cup \mathcal{A}_\mathcal{T})$ satisfying Properties 1-3, consider the one $\overline{\mathcal{D}}_\mathcal{T} = (\mathcal{N}_\mathcal{T}, \overline{\mathcal{H}}_\mathcal{T} \cup \overline{\mathcal{A}}_\mathcal{T})$ that also satisfies Property 4. We have that $z_R(\overline{\mathcal{D}}_\mathcal{T}) \ge z_R(\mathcal{D}_\mathcal{T})$ for all $D_\mathcal{T}$ satisfying Properties 1-3.*

Proof. The proof is provided in §EC.1.3 in the e-companion to this paper. □

We also include two additional sets of valid inequalities to further strengthen relaxation SND-HC-R($\mathcal{D}_\mathcal{T}$). We first observe that since variables w are nonnegative, equations (23) thus imply that

$$\sum_{((i,t),(j,\bar{t})) \in \mathcal{A}_\mathcal{T}} \tau_{ij} x_{ij}^{kt\bar{t}} \le l^k - e^k, \quad \forall\ k \in \mathcal{K}, \tag{24}$$

that is, a path P^k for commodity $k \in \mathcal{K}$ from the origin o^k to the destination d^k cannot exceed the maximum transit time computed as $l^k - e^k$. Define $\mathcal{K}_{ij}^k = \{k' : e^k + \phi^k(o^k, i) + \tau_{ij} + \phi^{k'}(j, d^{k'}) > l^{k'}$ or $e^{k'} + \phi^{k'}(o^{k'}, i) + \tau_{ij} + \phi^k(j, d^k) > l^k\}$. We observe that any commodity $k' \in \mathcal{K}_{ij}^k$ cannot be consolidated with commodity k on arc (i,j), thus deriving the following valid inequalities:

$$\left\lceil \frac{q^k}{u_{ij}} \right\rceil u_{ij} x_{ij}^{kt\bar{t}} + \sum_{k \in \mathcal{K}_{ij}^k} q^k x_{ij}^{kt\bar{t}} \le u_{ij} y_{ij}^{t\bar{t}}, \quad \forall\ ((i,t),(j,\bar{t})) \in \mathcal{A}_\mathcal{T}, k \in \mathcal{K}. \tag{25}$$

Inequalities (25) strengthen the following inequalities proposed by Marshall et al. (2021):

$$\left\lceil \frac{q^k}{u_{ij}} \right\rceil x_{ij}^{kt\bar{t}} \le y_{ij}^{t\bar{t}}, \quad \forall\ ((i,t),(j,\bar{t})) \in \mathcal{A}_\mathcal{T}, k \in \mathcal{K}. \tag{26}$$

Moreover, some variables could be safely removed to strengthen the relaxation further because they cannot be part of any optimal SND-HC-R($\mathcal{D}_\mathcal{T}$) solution. Following Marshall et al. (2021), variables $x_{ij}^{kt\bar{t}}$ can be safely removed if $t+\tau_{ij}+\phi^k(j,d^k)>l^k$ or there exists a time point $t'\in\mathcal{T}_i$ with $t'>t$ and $t'\le e^k+\phi^k(o^k,i)$. After removing these variables, for a time point $(i,t)\in\mathcal{N}_\mathcal{T}$, if there does not exist any variable $x_{ji}^{k\bar{t}t'}$ for an arc $((j,\bar{t}),(i,t'))\in\mathcal{A}_\mathcal{T}$ with $t'\in\mathcal{T}_i$ and $t'\le t$, according to the flow balance constraints, all variables $x_{ij}^{kt\bar{t}}$ for arcs $((i,t),(j,\bar{t}))\in\mathcal{A}_\mathcal{T}$ must take a value of zero and can be safely removed.

4.2. Initial partially time-expanded network

Without loss of generality, we assume that $min_{k\in\mathcal{K}}\{e^k\}=0$. The initial partially time-expanded network $\mathcal{D}_\mathcal{T}=(\mathcal{N}_\mathcal{T},\mathcal{H}_\mathcal{T}\cup\mathcal{A}_\mathcal{T})$ is defined in order to satisfy Properties 1-4 as follows:

- According to Property 1, for all $k\in\mathcal{K}$, node (o^k,e^k) and (d^k,l^k) are included in $\mathcal{N}_\mathcal{T}$.
- According to Properties 2 and 3, for each $i\in\mathcal{N}$, a node $(i,0)$ is added to $\mathcal{N}_\mathcal{T}$. Moreover, based also on Property 4, for each node $(i,t)\in\mathcal{N}_\mathcal{T}$ and for each arc $(i,j)\in\mathcal{A}$, arc $((i,t),(j,t'))$ with $t'=\arg\max\{s\in\mathcal{T}_j : s\le t+\tau_{ij}\}$ is added to $\mathcal{A}_\mathcal{T}$.
- For each $(i,t)\in\mathcal{N}_\mathcal{T}$, arc $((i,t),(i,t'))$ is added to $\mathcal{H}_\mathcal{T}$ where $t'=\arg\min\{s\in\mathcal{T}_i : s>t\}$, i.e., the holding arcs between two consecutive time points are added to set $\mathcal{H}_\mathcal{T}$.

Moreover, functions $\psi(.)$, $\vartheta(.)$, $\delta(.)$ and $\xi(.)$ are initialized based on the initial partially time-expanded network and expressions (14), (15), (18), and (17).

4.3. Computing a feasible CTSNDP-HC solution

Our exact algorithm utilizes a heuristic method based on the flat solution $\mathcal{S}=(\mathcal{P},\mathcal{C})$ computed by solving relaxation SND-HC-R($\mathcal{D}_\mathcal{T}$) and the implementable model $IM(\mathcal{S})$ described in §3.3 to derive a feasible CTSNDP-HC solution.

For a given flat solution $\mathcal{S}$ associated with a feasible solution of relaxation SND-HC-R($\mathcal{D}_\mathcal{T}$), due to inequalities (24), formulation $IM(\mathcal{S})$ without consolidation constraints (12d) can be decomposed into $|\mathcal{K}|$ subproblems, and always admits a feasible solution. Hence, the source of an infeasibility for formulation $IM(\mathcal{S})$ is related to consolidation constraints (12d). Based on this observation, we propose a heuristic method which iteratively solves formulation $IM(\mathcal{S})$ and selectively removes infeasible consolidation constraints (12d) when formulation $IM(\mathcal{S})$ is infeasible.

The heuristic method identifies infeasible consolidation constraints by computing an irreducibly inconsistent system (IIS) of formulation $IM(\mathcal{S})$, which is a description of the minimal subproblem that is still infeasible (Van Loon 1981, Chinneck and Dravnieks 1991). An infeasible subproblem is minimal if, when any of the constraints are removed, the infeasibility vanishes, and algorithms for identifying IIS have been investigated in Gleeson and Ryan (1990) and Chinneck (1997). As when an infeasible consolidation constraint (12d) for commodity k on arc (i,j) is removed from

Algorithm 2: Deriving a primal solution to the CTSNDP-HC

Input: Flat solution $\mathcal{S} = (\mathcal{P}, \mathcal{C})$
Output: Upper bound UB
begin
  `// Initialization`
  $\mathcal{J} \leftarrow \{1, \ldots, |\mathcal{C}|\}$;
  `// Derive a feasible CTSNDP-HC solution`
  Let LP be model $IM(\mathcal{S})$ without constraints (12d);
  Solve problem LP with consolidation constraints in $\mathcal{J}$;
  while *LP is infeasible* **do**
    Detect an IIS and select from the IIS the set $\mathcal{J}' \subseteq \mathcal{J}$ and compute $r^* = \arg\min\{\hat{f}_r : r \in \mathcal{J}'\}$;
    $\mathcal{J} \leftarrow \mathcal{J} \setminus \{r^*\}$;
    Solve problem LP with consolidation constraints in $\mathcal{J}$;
  end
  `// A feasible solution has been identified`
  $UB \leftarrow \hat{z}_{fc} + \hat{z}_w$;
  return UB;
end

the $IM(\mathcal{S})$, the total fixed cost may increase by $\hat{f} = f_{ij} \lceil q^k / u_{ij} \rceil$. Based on the latter observation, the heuristic identifies the infeasible consolidation constraint (12d) that has minimal $\hat{f}$. Algorithm 2 gives the steps of the heuristic method. In the algorithm, sets $\mathcal{J}$ represents index sets associated with consolidation constraints (12d). The algorithm iteratively solves an LP model until a feasible solution is found. The solution of the final LP model provides the arrival and departure times associated with the final feasible CTSNDP-HC solution of cost UB.

4.4. Refining a partially time-expanded network

This section describes the refinement strategies used to refine a partially time-expanded network. Our refinement strategies use the same refinement operations proposed by Boland et al. (2017) and also used by Hewitt (2022) to update the partially time-expanded network once new time points have been identified. They differ from the refinement strategies of Boland et al. (2017) and Hewitt (2022) on the methods used to determine the new time points.

Let (x, y, w) be an optimal solution of relaxation SND-HC-R($\mathcal{D}_\mathcal{T}$) with objective value LB, and let $\mathcal{S}$ be the corresponding flat solution. We have $LB = z_{fc}(\mathcal{S}) + z_w(LB)$, where $z_w(LB) = \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}} (h_i^k q^k) w_i^k$. Algorithm 2 computes an upper bound $UB = \hat{z}_{fc} + \hat{z}_w$. For the CTSNDP-HC, due to the presence of the in-storage holding costs and the approximation introduced by relaxation SND-HC-R($\mathcal{D}_\mathcal{T}$), an implementable solution $\mathcal{S}$ can correspond to a feasible solution with cost $UB = z_{fc}(\mathcal{S}) + z_w(\mathcal{S})$ greater than the cost of the current lower bound LB, and optimality cannot be proved. However, clearly, if $UB = LB$, then an optimal CTSNDP-HC solution has been identified. The following Lemma 1 gives optimality conditions under which a lower bound solution (x, y, w) of cost LB represents an optimal solution to the CTSNDP-HC such that $LB = UB$.

Lemma 1 *Consider an optimal solution (x,y,w) of relaxation SND-HC-R($\mathcal{D}_\mathcal{T}$) with cost LB. Define a solution θ_i^k, $i \in \mathcal{N}$, $k \in \mathcal{K}$, associated with solution (x,y,w), computed as follows:*

$$\theta_i^k = \begin{cases} \sum_{a=((i,t),(j,\bar{t}))\in\mathcal{A}_\mathcal{T}} t x_{ij}^{kt\bar{t}} - e^k, & i = o^k, \\ l^k - \sum_{a=((j,\bar{t}),(i,t))\in\mathcal{A}_\mathcal{T}} \psi^k(a)\, x_{ji}^{k\bar{t}t}, & i = d^k, \\ \sum_{a=((i,t),(j,\bar{t}))\in\mathcal{A}_\mathcal{T}} t x_{ij}^{kt\bar{t}} - \sum_{a=((j,\bar{t}),(i,t))\in\mathcal{A}_\mathcal{T}} \psi^k(a)\, x_{ji}^{k\bar{t}t}, & \textit{otherwise}. \end{cases} \quad \forall\, i \in \mathcal{N},\ k \in \mathcal{K}, \tag{27}$$

If (i) $\bar{t} = t + \tau_{ij}$ for each $a = ((i,t),(j,\bar{t})) \in \mathcal{A}_\mathcal{T}$ with $x_{ij}^{kt\bar{t}} = 1$ for some $k \in \mathcal{K}$ and (ii) $w_i^k = \theta_i^k$ for all $i \in \mathcal{N}$, $k \in \mathcal{K}$, then solution (x,y,w) is an optimal CTSNDP-HC solution of cost $UB = LB$.

Proof. The proof is provided in §EC.1.4 in the e-companion to this paper. □

If our exact algorithm (adapted from Algorithm 1 in §4) does not terminate, then the upper bound UB computed by Algorithm 2 is greater than the current lower bound LB, and the corresponding gap is greater than the given optimality tolerance. In this case, at least one of the optimality conditions in Lemma 1 is violated and the algorithm updates the partial time-expanded network $\mathcal{D}_\mathcal{T}$ with new time points (i,t) based on the following two cases.

(i) Some commodity $k \in \mathcal{K}$ passes through a short-arc $((i,t),(j,t')) \in \mathcal{A}$ with $t' < t + \tau_{ij}$. In this case, the flat solution $\mathcal{S}$ may be non-implementable, as there may be at least one commodity $k \in \mathcal{K}$ routed on a short-arc $((i,t),(j,t')) \in \mathcal{A}$ with $t' < t + \tau_{ij}$ that enables it to be involved in an infeasible consolidation. Even though the flat solution $\mathcal{S}$ is proven to be implementable, it may still contain some short-arcs. This implies that different from the case with zero holding costs, in which short-arcs need to be refined only when the flat solution $\mathcal{S}$ is non-implementable, the algorithm for the CTSNDP-HC needs to lengthen the short-arcs identified by adding new time points to network $\mathcal{D}_\mathcal{T}$ regardless the flat solution $\mathcal{S}$ is implementable or not. To do so, when the flat solution $\mathcal{S}$ is non-implementable, we apply the lengthen-arc procedure presented in Boland et al. (2017) for all short-arcs $((i,t),(j,t')) \in \mathcal{A}_\mathcal{T}$ with $t' < t + \tau_{ij}$ included in each IIS obtained by Algorithm 2. The lengthen-arc procedure lengthens the short-arc $((i,t),(j,t'))$ by invoking the *refine* and *restore* procedures presented in Boland et al. (2017) to add the time point $(i, t + \tau_{ij})$ to the node set $\mathcal{N}_\mathcal{T}$ and revise the corresponding arcs to satisfy the network properties. In addition, we refine an additional set of short-arcs $((i,t),(j,t')) \in \mathcal{A}_\mathcal{T}$ with $t' < t + \tau_{ij}$ used in the LB solution by also using the lengthen-arc procedure presented in Boland et al. (2017). We call the resulting refinement strategy *refinement strategy 1*.

(ii) There is at least one pair (i,k), $i \in \mathcal{N}$, $k \in \mathcal{K}$, such that $\theta_i^k \neq w_i^k$. We observe that when no commodity uses a short-arc $((i,t),(j,t'))$ with $t' < t + \tau_{ij}$, for each commodity $k \in \mathcal{K}$ the following equation holds

$$\sum_{i\in\mathcal{N}} \theta_i^k = l^l - e^k - \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_\mathcal{T}} (\bar{t} - t) x_{ij}^{k\,t\,\bar{t}} = l^l - e^k - \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_\mathcal{T}} \tau_{ij} x_{ij}^{kt\bar{t}} = \sum_{i\in\mathcal{N}} w_i^k, \tag{28}$$

as $\psi^k(a) = \bar{t}$ and $\bar{t} - t = \tau_{ij}$ for each $a = ((i,t),(j,\bar{t})) \in \mathcal{A}_\mathcal{T}$ with $x_{ij}^{kt\bar{t}} = 1$. The equation (28) illustrates that when the lower bound solution contains no short-arcs $((i,t),(j,t')) \in \mathcal{A}$ with $x_{ij}^{kt\bar{t}} = 1$ for some $k \in \mathcal{K}$, if $\theta \neq w$, there is at least one θ_i^k with $\theta_i^k \geq 0$ and $\theta_i^k < w_i^k$. The corresponding $\xi^k(a)$ is thus too weak to obtain a tight holding cost relaxation, with $w_i^k - \theta_i^k$ identifying the degree of the weakness of the value of $\xi^k(a)$. This implies that the value of $\vec{t}_i(t)$ should be reduced. Based on this observation, in this case, we only consider refining when we find a $\theta_i^k \geq 0$ with $\theta_i^k < w_i^k$ as finally all short-arcs will be eliminated by the case (i). For each $a = ((i,t),(j,\bar{t})) \in \mathcal{A}_\mathcal{T}$ with $x_{ij}^{kt\bar{t}} = 1$, $\theta_i^k \geq 0$ and $\theta_i^k < w_i^k$, we have $t + (w_i^k - \theta_i^k) \leq \vec{t}_i(t)$ as $w_i^k \leq \vec{t}_i(t) - \psi^k(a)$ and $\theta_i^k = t - \psi^k(a)$ according to (21) and (27), and $\vec{t}_i(t) > t + 1$ according to the definition of function $\vec{t}_i(t)$ in (16). We then identify the half-way time point between t and $t + (w_i^k - \theta_i^k)$ for node i, i.e., $(i, t + \max\{\lfloor (w_i^k - \theta_i^k)/2 \rfloor, 1\})$, which must be missing in the current network $\mathcal{D}_\mathcal{T}$ as $t < t + \max\{\lfloor (w_i^k - \theta_i^k)/2 \rfloor, 1\} < \vec{t}_i(t)$. In this way, we can generate a set of new time points $\mathcal{T}_{(i,t)}^{new}$ that aim to tighten the value of $\vec{t}_i(t)$ for each time point $(i,t) \in \mathcal{N}_\mathcal{T}$. We then add each new time point identified to the network $\mathcal{D}_\mathcal{T}$ by applying the *refine* and *restore* procedures presented in Boland et al. (2017). We call the resulting refinement strategy *refinement strategy 2*.

To summarise, to refine a partial network, we first apply refinement strategy 1 to refine short-arcs used in the LB solution and then apply refinement strategy 2 to further tighten the relaxation of holding times. The refine and restore procedures presented in Boland et al. (2017) and used in each refinement strategy will ensure that after refining, the four properties of the partial network are maintained. For refinement strategy 1, we found computationally convenient to lengthen the short-arcs identified by Algorithm 2 and lengthen a subset of the remaining short-arcs for the additional set. More specifically, after refining all short-arcs identified by Algorithm 2, the remaining short-arcs are first sorted for increasing order of their dispatch times and then at most $|\mathcal{K}|/5$ arcs are selected and lengthened. For refinement strategy 2, the time points (i, t^{new}) in $\mathcal{T}_{(i,t)}^{new}$ are added to the node set $\mathcal{N}_\mathcal{T}$ for increasing values of the time t^{new}. A new time point (i, t^{new}) is added only if the value t^{new} is less than the current $\vec{t}_i(t)$, as if $t^{new} \geq \vec{t}_i(t)$, the value of $\vec{t}_i(t)$ cannot be further reduced. It is worth noting that the four functions $\psi(.)$, $\vartheta(.)$, $\delta(.)$ and $\xi(.)$ change according to the updated network $\mathcal{D}_\mathcal{T}$.

4.5. Convergence and optimality

According to Lemma 1, in each iteration of the algorithm, if $LB < UB$ then at least one of optimality conditions (i) and (ii) is violated. Thus, according to the refinement strategy, at least one short-arc is lengthened, or at least one missing time point is added to the partial network. Since there are only a finite number of arcs and time points in the fully time-expanded network $\mathcal{D}_{\hat{\mathcal{T}}}$, after a finite number of iterations, the algorithm converges according to Lemma 1 or the network $\mathcal{D}_{\mathcal{T}}$ is equivalent to $\mathcal{D}_{\hat{\mathcal{T}}}$. In both cases, the algorithm will terminate with an optimal solution to the CTSNDP-HC. We have thus established the following result.

Theorem 3 *If the optimality tolerance is set equal to zero, the DDD algorithm for the CTSNDP-HC converges to optimality after finitely many iterations.*

5. Computational experiments

In this section, we present extensive computational analysis based on two sets of experiments. Through the first set of experiments (see §5.1), we evaluate the performance of the DDD algorithm for the CTSNDP-HC on instances derived from the literature. Through the second set of experiments (see §5.2), we examine the quality of the solutions produced on an additional set of instances to analyze the impact of holding costs on the decisions in CTSNDP. Based on the results from these two sets of experiments, we evaluate the effectiveness of the proposed DDD algorithm for the CTSNDP-HC and investigate the importance and benefits of taking holding costs into account in solving the CTSNDP.

We denote with EXM our DDD algorithm for the CTSNDP-HC. In order to further compare the performance of EXM, we also implemented the algorithm proposed by Boland et al. (2017) for the CTSNDP, hereafter denoted as EXM-0. A preliminary experiment can be found in §EC.2.4 of the e-companion, which confirms the effectiveness of our implementation of the EXM-0 algorithm. Thus, we take the algorithm EXM-0 as a baseline algorithm to compute heuristic solutions for the CTSNDP-HC and to analyze the performance of the algorithm EXM. In the following experiments, we solve each of the considered instances to an optimility gap of 0.01 with a time limit of 2 hours for both EXM and EXM-0. More implementation details of computational experiments, including the coding and computational environment, can be found in EC.2.1 of the e-companion.

5.1. Experiments based on CTSNDP-HC instances generated from CTSNDP benchmark instances

In our first set of experiments, we aimed to test the performance of EXM in solving CTSNDP benchmark instances used in the literature and CTSNDP-HC instances generated from these CTSNDP benchmark instances.

5.1.1. Instances We considered the set of 558 CTSNDP instances generated by Boland et al. (2018) that also contains the 432 instances used by Boland et al. (2017). The 558 instances were also used by Marshall et al. (2021) to obtain their computational results. These instances were derived from 31 classes of the "C" instances presented in Crainic et al. (2001) which have been used as benchmark instances for the capacitated fixed charge network design problem. For each of the 31 classes of networks, Boland et al. (2018) generated 18 CTSNDP timed instances by first calculating the travel times for each arc and then by generating the time windows for each commodity by randomly sampling from a normal distribution. Based on Boland and Savelsbergh (2019) and as also reported by Marshall et al. (2021), these instances can also be grouped by the flexibility and cost ratio of the instances, these being a measure of the tractability of the instances. An instance has (i) *low flexibility* (LF) if $\min_{k\in\mathcal{K}}\{l^k - (e^k + \underline{\tau}_{o^k d^k})\} < 227$, and *high flexibility* (HF) otherwise, and (ii) *low cost* ratio (LC) if $\frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} \frac{f_a}{c_a u_a} < 0.175$, and *high cost* ratio (HC) otherwise. The instances are then grouped according to the two measures, resulting in the four groups of instances, namely, "HC/LF", "HC/HF", "LC/LF" and "LC/HF". For each of the considered CTSNDP instances, we generated a CTSNDP-HC instance by setting the per-unit-of-demand-and-time holding cost h_i^k (see §EC.2.2 of the e-companion for details).

5.1.2. Results In this section, we analyze the performance of EXM in solving instances of both CTSNDP (with zero holding costs) and CTSNDP-HC (with nonzero holding costs). We executed EXM for the corresponding 558 CTSNDP-HC instances and executed EXM-0 for the corresponding 558 CTSNDP instances. To attest to the effectiveness of EXM in solving the CTSNDP instances, we also used algorithm EXM to solve the set of CTSNDP instances, that is, EXM is used by setting the holding costs equal to zero.

Table 2 reports the corresponding results based on the categories of the flexibility and cost ratio of the instances. The following notation is used:

- $LB0$, $UB0$: final lower and upper bounds computed by EXM-0, respectively.
- $UB1$: value of the CTSNDP-HC solution derived from the upper bound $UB0$ computed by algorithm EXM-0, calculated based on the flat solution $\hat{\mathcal{S}}$ of the solution $UB0$. Upper bound $UB1$ is computed as cost $z_{fc}(\hat{\mathcal{S}}) + z_w(\hat{\mathcal{S}})$ where $z_w(\hat{\mathcal{S}})$ is the the optimal solution cost of problem $IM(\hat{\mathcal{S}})$.
- LB, UB: final lower and upper bounds computed by EXM, respectively.

For each method and group of instances, the table shows the percentage of instances solved to optimality ("%*opt*"), the average percentage deviation of lower bound LB with respect to lower bound $LB0$ (i.e., $\%LB0 = 100.0 \times \frac{LB-LB0}{LB0}$) and the percentage deviations of the different upper bounds computed as $\%UB0 = 100.0 \times \frac{UB0-LB0}{UB0}$, $\%UB1 = 100.0 \times \frac{UB1-UB}{UB1}$ and $\%UB = 100.0 \times \frac{UB-LB}{UB}$. For the different percentage deviations, the *min* and *max* values are also reported. For $UB0$ and UB, the deviations are computed over all instances not solved to optimality, whereas the deviations of $UB1$ are computed over all instances. Columns *time* and *iter* give the average computing time and number of iterations, respectively, computed over all instances. Column $\%tLB$ reports the percentage of the total time spent in computing the lower bound, i.e., the percentage of the total time spent by the MIP solver over the total computing time.

Table 2 Summary results on the CTSNDP-HC instances

	Zero holding costs														Nonzero holding costs										
	EXM-0							EXM							EXM										
		%UB0							%UB							%UB						%LB0	%UB1	
Group	*%opt*	*min*	*max*	*avg*	*time*	*%tLB*	*iter*	*%opt*	*min*	*max*	*avg*	*time*	*%tLB*	*iter*	*%opt*	*min*	*max*	*avg*	*time*	*%tLB*	*iter*	*avg*	*avg*	*max*
HC/LF	96.7	1.1	4.5	3.3	345.3	90.4	10.1	99.5	1.1	1.1	1.1	181.8	86.0	4.2	98.4	1.1	1.9	1.6	279.3	86.6	4.4	4.1	0.8	5.3
HC/HF	71.8	1.0	23.3	7.5	2726.1	96.3	11.6	81.9	1.1	5.0	2.4	1936.0	93.6	6.5	65.5	1.0	6.1	2.9	2902.7	94.4	6.2	10.6	3.9	17.9
LC/LF	100.0	-	-	-	0.5	66.6	3.5	100.0	-	-	-	0.6	68.6	1.8	100.0	-	-	-	0.7	62.3	1.8	0.7	0.0	1.0
LC/HF	100.0	-	-	-	0.1	52.8	1.6	100.0	-	-	-	0.1	68.6	1.2	100.0	-	-	-	0.2	57.0	2.3	0.8	1.1	8.0

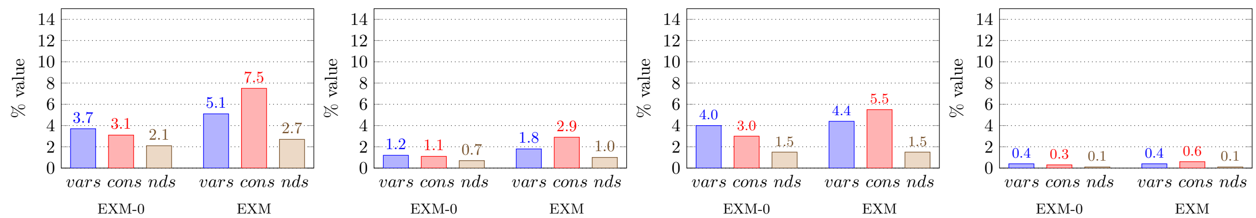
Note: "-" represents that all instances in the corresponding instance group are solved to optimal by the corresponding method.

Table 2 indicates that EXM achieves a better performance than that of EXM-0 when used to solve the CTSNDP instances with zero holding cost. Algorithm EXM can solve more instances to optimality and provides better optimality gaps for the unsolved instances. Moreover, on average, it requires a smaller number of iterations and a shorter computing time to converge to optimality. This improved performance can be due to the effectiveness of the new valid inequalities (25), as shown by Table 4 for algorithm EXM.

The results on the CTSNDP-HC instances show that CTSNDP-HC instances of groups **LC/LF** and **LC/HF** can also be easily solved by EXM, as with the CTSNDP case. Conversely, the instances of groups **HC/LF** and **HC/HF** are more difficult to solve, as shown by the percentages of instances solved to optimality. In particular, for groups **HC/HF** that are characterized by a high cost ratio and high flexibility, a trade-off between fixed, flow, and holding costs is the most difficult to achieve.

The summary results of Table 2 show that, compared with EXM-0, algorithm EXM generates better lower and upper bounds for the CTSNDP-HC. On the most difficult instances of group **HC/HF**, the solutions obtained by EXM improve those derived from the solutions obtained by EXM-0 significantly. It thus implies that a significant cost saving can be gained in some cases by taking into account holding costs when solving the CTSNDP. A more significant improvement on the lower bound is observed for the instances with a high cost ratio. Not surprisingly, for the **LC/LF** instances, the upper bound $UB1$ based on EXM-0 has comparable performance to the solutions obtained by EXM, as the low cost ratio and low flexibility leave little room for improvement on the total cost by adjusting the holding plan.

To further analyze the effectiveness of the DDD approach and impact of the holding cost, Figure 4 shows the relative sizes (average percentage values), in terms of the number of variables ("$vars$") and constraints ("*cons*") of the final relaxation models (models SND($\mathcal{D}_\mathcal{T}$) and SND-HC-R($\mathcal{D}_\mathcal{T}$) associated with the last iteration of the DDD algorithm) solved by algorithms EXM-0 and EXM with respect to the models associated with the fully time-expanded networks. The figure also shows the relative number of time points ("$\%nds$") of the final partial network $\mathcal{D}_\mathcal{T}$ over the full network.



(a) Group **HC/LF** (b) Group **HC/HF** (c) Group **LC/LF** (d) Group **LC/HF**

Figure 4 Comparison of partially and fully time-expanded networks

The figure clearly shows the advantage of the DDD approach, which is capable of computing optimal solutions considering only a reduced set of time points of the full network. What is more, the sizes of the MIP solvers solved at the different iterations are confined to small portions of the model associated with the fully time-expanded networks, a very relevant feature given the complexity of solving time-index formulations. A comparison with the results using EXM-0 shows that EXM requires about twice the number of variables and constraints of EXM-0 for some instances set, which indicates an increased complexity of the problem after incorporating holding costs.

Table 3 CTSNDP-HC instances: holding costs, holding times and consolidations

	HC/LF			**HC/HF**			**LC/LF**			**LC/HF**		
	$\%hc$	$\%ht$	$\%cs$	$\%hc$	$\%ht$	$\%cs$	$\%hc$	$\%ht$	$\%cs$	$\%hc$	$\%ht$	$\%cs$
UB1	4.4	7.8	46.9	9.2	9.9	55.4	0.6	3.2	14.7	1.9	2.4	9.5
UB	2.9	6.0	44.1	5.6	7.1	52.2	0.4	2.5	12.6	0.5	1.0	5.7

Table 3 summarizes relevant details of the solutions corresponding to the upper bounds $UB1$ and UB for the instances that are solved to optimality by both EXM and EXM-0. More specifically, the table reports the following average percentage values: (i) $\%hc$, the holding cost over the total cost, (ii) $\%ht$, the holding time over the total transit time, and (iii) $\%cs$, the number of consolidation arcs over the total number of arcs used by the solution. The table also shows that the solutions computed by EXM achieve a marginal reduction in terms of holding times (and corresponding holding costs) with respect to the solutions derived from EXM-0. This reduction is achieved by a decrease in the percentages of consolidations (see column $\%cs$). These also reveal the significant effect of the holding cost on the decisions of holding and consolidation in the CTSNDP.

We conducted additional experiments to verify the effectiveness of each proposed component of EXM. To evaluate the effectiveness of valid inequalities (22) and (25), we tested the following variants of algorithm EXM: i) without constraints (22) and (25), ii) without constraints (25), and iii) replacing constraints (25) with constraints (26) described by Marshall et al. (2021). During preliminary experiments, we observed that the effectiveness of the valid inequalities is negligible in the LC class of instances. We, therefore, give the results on the HC class of instances using the

Table 4 Computational results for EXM variants with different valid inequalities

Group	Algorithm	$\%LB_{fi}$	$\%UB$			*time*	*iter*	*%opt*
			min	*max*	*avg*			
HC/LF	EXM - (22) - (25)	0.0	1.3	2.5	1.9	824.1	8.1	95.6
	EXM - (25)	1.7	1.1	2.4	1.6	572.2	6.0	95.6
	EXM - (25) + (26)	3.0	1.1	1.6	1.3	295.2	4.8	98.4
	EXM	5.6	1.1	1.9	1.6	279.3	4.4	98.4
HC/HF	EXM - (22) - (25)	0.0	1.0	13.4	3.9	3444.7	8.6	56.5
	EXM - (25)	4.3	1.0	10.0	3.7	3279.0	6.6	60.5
	EXM - (25) + (26)	5.1	1.0	9.3	3.2	3055.4	6.5	62.7
	EXM	10.2	1.0	6.1	2.9	2902.7	6.2	65.5

same notation introduced in Table 2. Table 4 shows the results obtained where column $\%LB_{fi}$ reports the average percentage deviation of the lower bound LB obtained at the first iteration of each algorithm with respect to the lower bound obtained at first iteration by algorithm EXM without constraints (22) and (25). The table shows that constraints (22) and (25) significantly affect algorithm performance. Indeed, after adding constraints (22), 4% more HC/HF instances can be solved to optimality, and a tighter lower bound is achieved as shown by the $\%LB_{fi}$ values. The constraints (26) proposed by Marshall et al. (2021) improve the algorithm performance in terms of both the optimality rate and the computational time. However, using constraints (25), the algorithm can be further enhanced, as 3% more HC/HF instances are solved to optimality. In addition, the lower bound LB at the first iteration is tightened further. Section §EC.2.5 of the e-companion to this paper reports the results of the upper bound heuristic and refinement strategy.

5.2. Experiments on newly generated CTSNDP-HC benchmark instances

For our second set of experiments, we generated a new set of CTSNDP-HC instances with different levels of connectivity of the underlying physical network (spatial component) and flexibility of the shipments' time requirements (temporal component). We thus examine the benefits gained by incorporating holding costs into the CTSNDP and the impact of the holding cost on the solution structure in these more differentiated instances.

5.2.1. Instances We generated the new instances with different network connectivity and time flexibility based on 31 sets of the "C" instances presented in Crainic et al. (2001). We considered four levels of network connectivity, i.e., D_1, D_2, D_3, D_4, and three different time flexibilities drawn from three different normal contributions, i.e., distribution A, B and C, respectively. As such, we obtained 12 sets of new instances, for each of which, we randomly generated three instances, so that a total of $3 \times (31 \times 4 \times 3) = 1116$ instances were produced.

Given an instance and the associated network $\mathcal{D}_x$, $x \in \{1,2,3,4\}$, the minimum $o^k - d^k$ cut (denoted as $(S, \mathcal{N} \setminus S)_k$) in $\mathcal{D}_x$, $\forall k \in \mathcal{K}$ and $\min_{k\in\mathcal{K}}\{|(S, \mathcal{N} \setminus S)_k|\}$, i.e., the cardinality of the cut

Table 5 Connectivity properties of the new CTSNDP-HC instances

Network	Min cut.	Avg cut.
$\mathcal{D}_1$	7	12
$\mathcal{D}_2$	4	8
$\mathcal{D}_3$	2	4
$\mathcal{D}_4$	1	1

Table 6 Time flexibility of the new CTSNDP-HC instances

Normal Distribution	Mean(μ)	StdDev(σ)
A	$\frac{1}{2}l_{avg}$	$\frac{1}{6}\mu$
B	l_{avg}	$\frac{1}{6}\mu$
C	$\frac{3}{2}l_{avg}$	$\frac{1}{6}\mu$

having the minimum cardinality among the different $o^k - d^k$ pair, can be easily computed. For each type of network, Table 5 summarizes the connectivity properties of the newly generated CTSNDP-HC instances by reporting the cardinality of the cut having the minimum cardinality ("Min cut") and the average cardinality of all the minimum $o^k - d^k$ cuts ("Avg cut") computed over all instances belonging to this particular type of network. Norms and standard deviations of the three different normal distributions (named distribution A, B and C) are shown in Table 6, where l_{avg} represents the average length of the shortest-time paths over all commodities. For other details of the generation of these 1116 instances, see §EC.2.3 of the online companion.

5.2.2. Results For each of the newly generated CTSNDP-HC instances, we execute algorithm EXM and EXM-0, with the holding costs set to be zero for EXM-0. Table 7 summarizes the results obtained using the same notation introduced in Table 2. The instances are grouped by distribution and network types. The average values of *time*, and *iter* are computed over all instances. The deviations relative to $UB0$ and UB are computed over all instances not solved to optimality, whereas the deviations relative to $LB0$ and $UB1$ are computed over all instances.

EXM significantly outperforms EXM-0 in this set of instances regarding the optimality rate and computational time. Therefore, the results confirmed the effectiveness of our new relaxation tightening methods, upper bound heuristic, and refinement strategies. We also note that an increasing time flexibility corresponds to the ordering of distributions A, B and C, whereas a decreasing connectivity level is associated with the ordering of networks $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$ and $\mathcal{D}_4$. The results shown in the table indicate that, for both EXM-0 (CTSNDP case) and EXM (CTSNDP-HC case), the instances characterized by high time flexibility and connectivity levels are particularly difficult. Such difficulty is due to the following reason: Under higher time flexibility and connectivity, more consolidation opportunities are allowed, as well as more alternative services are available. These increase the difficulty of the MIP problems to be solved, displaying many equivalent solutions that significantly blows up the sizes of the branch-and-bound trees. As a consequence, and as also shown by the percentage of the time spent in solving the relaxation model, the entire solution process is slowed down.

Table 7 Summary results on the new CTSNDP-HC instances

		Zero holding costs							Nonzero holding costs									
		EXM-0							EXM									
			$\%UB0$							$\%UB$						$\%LB0$	$\%UB1$	
Dist.	Network	$\%opt$	min	max	avg	$time$	$\%tLB$	$iter$	$\%opt$	min	max	avg	$time$	$\%tLB$	$iter$	avg	avg	max
A	$\mathcal{D}_1$	86.0	1.1	18.0	8.6	1394.2	82.5	7.0	97.8	1.1	2.1	1.6	183.7	86.3	4.0	7.3	4.1	15.7
	$\mathcal{D}_2$	83.9	1.1	21.2	8.2	1439.5	80.4	5.8	98.9	2.3	2.3	2.3	116.7	84.9	3.8	8.0	4.6	17.6
	$\mathcal{D}_3$	86.0	1.7	33.0	11.2	1286.8	78.4	5.7	98.9	1.2	1.2	1.2	175.0	80.5	3.8	8.8	5.2	24.4
	$\mathcal{D}_4$	87.1	2.2	24.5	9.7	1001.9	76.9	5.8	98.9	1.8	1.8	1.8	103.0	78.6	3.8	9.0	5.9	23.4
B	$\mathcal{D}_1$	58.1	2.8	46.7	19.0	3293.3	90.2	3.8	84.9	1.1	6.8	3.0	1810.4	95.5	4.7	10.0	7.7	36.5
	$\mathcal{D}_2$	58.1	3.4	49.1	19.1	3316.5	88.7	3.3	90.3	1.3	7.7	4.0	1246.9	94.6	4.8	11.8	9.9	34.8
	$\mathcal{D}_3$	54.8	1.3	47.5	14.3	3429.4	82.5	2.8	89.2	1.0	6.3	2.7	906.9	89.8	4.9	11.3	10.1	37.1
	$\mathcal{D}_4$	76.3	2.0	44.0	15.5	1947.9	79.3	4.1	95.7	1.3	2.8	1.9	348.2	85.1	5.1	10.7	10.4	37.1
C	$\mathcal{D}_1$	47.3	1.5	57.2	22.7	3920.0	93.4	3.3	53.8	1.0	25.4	8.7	3586.5	98.5	4.0	6.9	6.5	44.7
	$\mathcal{D}_2$	48.4	1.1	56.2	19.3	3838.7	90.6	2.9	59.1	1.0	19.7	4.6	3288.5	97.4	4.7	8.1	8.6	46.1
	$\mathcal{D}_3$	52.7	3.7	56.1	16.2	3589.4	86.3	2.7	84.9	1.0	19.9	4.6	1650.8	94.0	5.1	11.0	11.8	43.1
	$\mathcal{D}_4$	68.8	1.6	35.4	12.6	2372.3	81.1	3.4	94.6	3.0	3.0	3.0	452.3	87.5	5.6	10.6	14.1	45.8

In order to analyze the impact of the holding cost on the holding and consolidation decisions of the CTSNDP, Table 8 reports a comparison between the solution structures of upper bounds UB and $UB1$. Since the newly generated CTSNDP-HC instances are also based on the 31 "C" classes of the CTSNDP benchmark instances, they can also be classified into 31 groups. For a fair comparison, we only compare the group of instances for which all instances were solved optimally by both EXM and EXM-0. The table shows the deviations of the upper bound value ($\%UB1$), the number of consolidations ($\%cs$), and the ratios of the holding costs ($\%hc$) and holding times ($\%ht$). In addition, the table gives the difference between the timed arcs used in the solutions computed as $da = \frac{1}{2} \times \left(\frac{|A(UB1)| - |A(UB1) \cap A(UB)|}{|A(UB1)|} + \frac{|A(UB)| - |A(UB1) \cap A(UB)|}{|A(UB)|} \right)$, where $A(UB)$ and $A(UB1)$ are the sets of timed arcs used in the solutions of UB and $UB1$, respectively. Table 8 reports the average of ratio da ($\%da$). Table 8 also presents, for upper bounds UB and $UB1$, the percentage of the commodities with different routing or scheduling plans ($\%dp$). In particular, the percentage of the commodities that use different delivery routes is reported under column $\%dr$, and the percentage of the commodities that use the same delivery routes but with different departure scheduling plans is reported under column $\%ds$. Columns "*avg*" and "*max*" give average and maximum values of corresponding ratios.

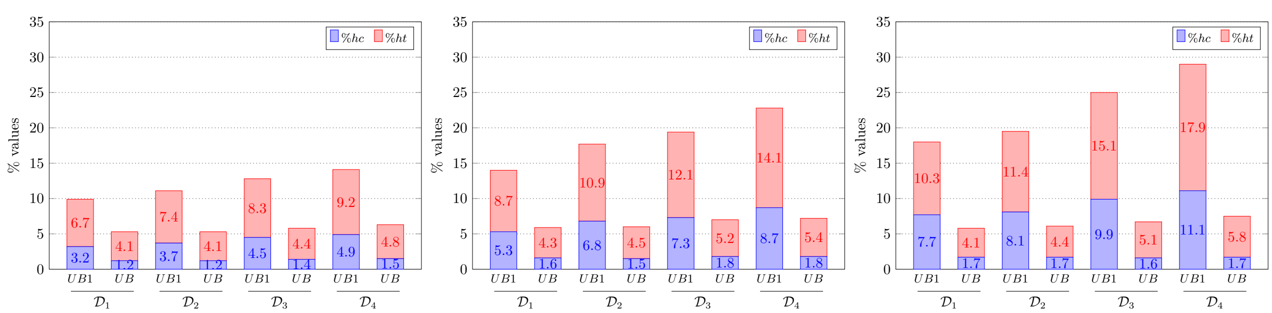
Table 8 shows significant improvements of the optimal cost UB computed by EXM with respect to the cost $UB1$ derived from the optimal CTSNDP solution computed by EXM-0. The improvements clearly show the impact of the holding cost on the cost of the final solution. They indicate that solving the CTSNDP and, based on the corresponding solutions, deriving corresponding CTSNDP-HC solutions is not a valid option. The table also shows that, on average, the solutions differ for about 23% of the total timed arcs used, and on average, one-third of

Table 8 **Analysis of the difference in terms of timed arcs used between upper bounds $UB1$ and UB**

Dist.	Network	$\%UB1$	$\%dp$	$\%da$	$\%dr$		$\%ds$		EXM-0			EXM		
					avg	max	avg	max	$\%hc$	$\%ht$	$\%cs$	$\%hc$	$\%ht$	$\%cs$
A	$\mathcal{D}_1$	1.6	22.4	14.1	1.0	7.5	21.4	42.5	3.2	6.7	40.9	1.2	3.9	32.9
	$\mathcal{D}_2$	2.0	25.7	17.0	1.0	7.0	24.8	38.0	3.7	7.4	43.7	1.2	4.1	35.2
	$\mathcal{D}_3$	2.4	29.8	19.9	0.4	2.6	29.4	42.5	4.5	8.3	49.1	1.4	4.4	38.0
	$\mathcal{D}_4$	2.7	34.0	22.8	0.0	0.0	34.0	56.0	4.9	9.2	52.1	1.5	4.8	40.6
B	$\mathcal{D}_1$	3.0	25.6	16.8	2.1	7.5	23.5	45.0	5.3	8.7	47.4	1.6	4.3	38.2
	$\mathcal{D}_2$	4.4	31.5	20.7	1.6	12.5	30.0	50.0	6.8	10.9	52.2	1.5	4.5	39.1
	$\mathcal{D}_3$	5.2	35.7	25.1	1.2	5.1	34.6	60.0	7.7	12.1	54.4	1.8	5.2	42.6
	$\mathcal{D}_4$	6.2	41.7	29.0	0.0	1.0	41.7	63.0	8.7	14.1	56.5	1.8	5.4	44.4
C	$\mathcal{D}_1$	5.1	31.2	20.9	3.3	11.0	27.9	44.0	7.7	10.3	50.6	1.7	4.1	37.2
	$\mathcal{D}_2$	5.5	34.9	24.3	3.0	12.5	31.9	50.0	8.1	11.4	54.9	1.7	4.4	40.0
	$\mathcal{D}_3$	7.3	39.7	26.2	1.6	10.0	38.1	57.5	9.9	15.1	58.0	1.6	5.1	41.5
	$\mathcal{D}_4$	8.7	47.6	34.0	0.0	0.0	47.6	72.5	11.1	17.9	61.5	1.7	5.8	45.6

commodities take different routing and departure plans. In particular, for the case of distribution C and network $\mathcal{D}_2$, up to 12.5% of the commodities are delivered by different paths. These differences also show a reduction in the rate of holding times, holding costs, and consolidation arcs of the solutions of UB with respect to the solutions of $UB1$. The differences in $\%UB1$, $\%dp$, and $\%da$ are more significant for increasing flexibility level, and for a fixed flexibility level, for decreasing connectivity level. The results demonstrate the holding costs' significant impact on the solution structure.

Figure 5 depicts the ratios of the holding costs and holding times shown in Table 8. Figure 5 shows significant increases in the two measures regarding upper bound $UB1$, again depending on the connectivity of the underlying physical network and the flexibility of the shipments' time requirements. The values of the two measures are pretty stable concerning UB under different levels of connectivity and flexibility, thus further highlighting the importance of considering holding costs in solving the CTSNDP-HC, especially for instances with high flexibility and lower connectivity.



(a) Distribution A (b) Distribution B (c) Distribution C

Figure 5 **Ratios of the holding costs and holding times**

Furthermore, we conducted a sensitivity analysis on the unit holding cost. The results show that the impact is significant for instances under different unit holding cost values. See §EC.2.6 of the e-companion for details.

6. Conclusions

In this paper, we designed a new exact algorithm for a generalization of the continuous-time service network design problem (CTSNDP), first studied by Boland et al. (2017), where in addition to fixed and flow costs, holding costs are also considered (CTSNDP-HC). The exact algorithm uses the same dynamic discretization discovery (DDD) solution framework proposed by Boland et al. (2017) for the CTSNDP, but it extends the DDD framework in a number of non-trivial ways by exploiting a new relaxation of a complete time-index model, a new upper bound heuristic, and a new refinement strategy. The new algorithm was extensively tested both on instances derived from the literature and on newly generated instances, with the aim of benchmarking the essential factors of the CTSNDP-HC. In particular, to assess the impact of the holding costs, we designed experiments by varying the connectivity of the underlying physical network and the flexibility of the shipments' time requirements. The results obtained not only show the effectiveness of the new exact algorithm in solving challenging CTSNDP-HC instances involving up to 400 commodities, but also indicate that ignoring the holding costs leads to poor quality solutions. The impact of holding costs is significant particularly when the network is characterized by high flexibility and low connectivity levels. Compared with the heuristic solution obtained by the method ignoring the holding cost, the maximum cost saving percentage of the minimal total cost achieved by our exact method can be up to 46.1%.

Finally, it is worth mentioning that the DDD algorithm for the CTSNDP proposed by Boland et al. (2017) has been recently improved by Marshall et al. (2021) by considering time-expanded networks based on time intervals instead of time points, and that the algorithm proposed in this paper can be tailored to consider time intervals, together with the holding costs. In addition, real-world service network design problems pose several challenging extensions, such as asset management, terminal capacities and compatibility of commodities, which are characterized by many shipments (and corresponding commodities). Our future work will therefore go in the direction of considering these extensions and other important features of this class of problems.

Acknowledgments

We gratefully acknowledge the help of Prof. Natashia Boland and Dr. Luke Marshall in providing us with problem instances and algorithm details of Boland et al. (2017) and Marshall et al. (2021).

The authors would like to thank the anonymous reviewers, associate editor and editor for their extremely helpful suggestions and very thorough review of the paper. In particular, we are extremely grateful to an anonymous referee for several helpful comments that helped improve the quality of our solution approach.

References

Andersen, J., Christiansen, M., Crainic, T. G. and Grønhaug, R. (2011), Branch and price for service network design with asset management constraints, *Transportation Science* **45**(1), 33–49.

Andersen, J., Crainic, T. G. and Christiansen, M. (2009*a*), Service network design with asset management: Formulations and comparative analyses, *Transportation Research Part C: Emerging Technologies* **17**(2), 197–207.

Andersen, J., Crainic, T. G. and Christiansen, M. (2009*b*), Service network design with management and coordination of multiple fleets, *European Journal of Operational Research* **193**(2), 377–389.

Belieres, S. (2019), Mathematical programming for tactical transportation planning in a multi-productsupply chain, PhD thesis, Automatic Control Engineering. INSA de Toulouse.

Bertsimas, D. and Weismantel, R. (2005), *Optimization over integers.*, Athena Scientific.

Boland, N., Hewitt, M., Marshall, L. and Savelsbergh, M. (2017), The continuous-time service network design problem, *Operations Research* **65**(5), 1303–1321.

Boland, N., Hewitt, M., Marshall, L. and Savelsbergh, M. (2018), The price of discretizing time: a study in service network design, *EURO Journal on Transportation and Logistics* pp. 1–22.

Boland, N., Kalinowski, T. and Kaur, S. (2015), Scheduling network maintenance jobs with release dates and deadlines to maximize total flow over time: Bounds and solution strategies, *Computers & Operations Research* **64**, 113–129.

Boland, N. L. and Savelsbergh, M. W. (2019), Perspectives on integer programming for time-dependent models, *TOP* pp. 1–27.

Bookbinder, J. H. and Higginson, J. K. (2002), Probabilistic modeling of freight consolidation by private carriage, *Transportation Research Part E: Logistics and Transportation Review* **38**(5), 305–318.

Chinneck, J. W. (1997), Finding a useful subset of constraints for analysis in an infeasible linear program, *INFORMS Journal on Computing* **9**(2), 164–174.

Chinneck, J. W. and Dravnieks, E. W. (1991), Locating minimal infeasible constraint sets in linear programs, *ORSA Journal on Computing* **3**(2), 157–168.

Crainic, T. G. (2000), Service network design in freight transportation, *European Journal of Operational Research* **122**(2), 272–288.

Crainic, T. G., Frangioni, A. and Gendron, B. (2001), Bundle-based relaxation methods for multicommodity capacitated fixed charge network design, *Discrete Applied Mathematics* **112**(1-3), 73–99.

Crainic, T. G., Hewitt, M., Toulouse, M. and Vu, D. M. (2014), Service network design with resource constraints, *Transportation Science* **50**(4), 1380–1393.

Crainic, T. G., Hewitt, M., Toulouse, M. and Vu, D. M. (2018), Scheduled service network design with resource acquisition and management, *EURO Journal on Transportation and Logistics* **7**(3), 277–309.

Crainic, T. G. and Rousseau, J.-M. (1986), Multicommodity, multimode freight transportation: A general modeling and algorithmic framework for the service network design problem, *Transportation Research Part B: Methodological* **20**(3), 225–242.

Dash, S., Günlük, O., Lodi, A. and Tramontani, A. (2012), A time bucket formulation for the traveling salesman problem with time windows, *INFORMS Journal on Computing* **24**(1), 132–147.

Erera, A., Hewitt, M., Savelsbergh, M. and Zhang, Y. (2013), Improved load plan design through integer programming based local search, *Transportation Science* **47**(3), 412–427.

Farvolden, J. M. and Powell, W. B. (1994), Subgradient methods for the service network design problem, *Transportation Science* **28**(3), 256–272.

Fleischer, L. and Skutella, M. (2007), Quickest flows over time, *SIAM Journal on Computing* **36**(6), 1600–1630.

Frangioni, A. and Gendron, B. (2009), 0–1 reformulations of the multicommodity capacitated network design problem, *Discrete Applied Mathematics* **157**(6), 1229–1241.

Gendron, B., Crainic, T. G. and Frangioni, A. (1999), Multicommodity capacitated network design, *in* B. Sansò and P. Soriano, eds, 'Telecommunications Network Planning', Centre for Research on Transportation, Springer US, Boston, MA, pp. 1–19.

Ghamlouche, I., Crainic, T. G. and Gendreau, M. (2003), Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design, *Operations Research* **51**(4), 655–667.

Gleeson, J. and Ryan, J. (1990), Identifying minimally infeasible subsystems of inequalities, *ORSA Journal on Computing* **2**(1), 61–63.

Groß, M. and Skutella, M. (2012), Maximum multicommodity flows over time without intermediate storage, *in* 'European Symposium on Algorithms', Springer, pp. 539–550.

Gurobi Optimization, L. (2021), 'Gurobi optimizer reference manual (v.8.1.1)'. http://www.gurobi.com.

Hewitt, M. (2019), Enhanced dynamic discretization discovery for the continuous time load plan design problem, *Transportation Science* **53**(6), 1731–1750.

Hewitt, M. (2022), The Flexible Scheduled Service Network Design Problem, *Transportation Science* **56**(4), 1000–1021.

Hosseininasab, A. (2015), The continuous time service network design problem, Master's thesis, University of Waterloo.

Hu, W., Toriello, A. and Dessouky, M. (2018), Integrated inventory routing and freight consolidation for perishable goods, *European Journal of Operational Research* **271**(2), 548–560.

Jarrah, A. I., Johnson, E. and Neubert, L. C. (2009), Large-scale, less-than-truckload service network design, *Operations Research* **57**(3), 609–625.

Lagos, F., Boland, N. and Savelsbergh, M. (2020), The continuous-time inventory-routing problem, *Transportation Science* **54**(2), 375–399.

Lai, M., Cai, X. and Hall, N. G. (2022), Cost allocation for less-than-truckload collaboration via shipper consortium, *Transportation Science* **56**(3), 585–611.

Marshall, L., Boland, N., Savelsbergh, M. and Hewitt, M. (2021), Interval-based dynamic discretization discovery for solving the continuous-time service network design problem, *Transportation Science* **55**(1), 29–51.

Medina, J., Hewitt, M., Lehuédé, F. and Péton, O. (2019), Integrating long-haul and local transportation planning: The service network design and routing problem, *EURO Journal on Transportation and Logistics* **8**(2), 119–145.

Pedersen, M. B., Crainic, T. G. and Madsen, O. B. (2009), Models and tabu search metaheuristics for service network design with asset-balance requirements, *Transportation Science* **43**(2), 158–177.

Rudi, A., Fröhling, M., Zimmer, K. and Schultmann, F. (2016), Freight transportation planning considering carbon emissions and in-transit holding costs: a capacitated multi-commodity network flow model, *EURO Journal on Transportation and Logistics* **5**(2), 123–160.

Skutella, M. (2009), An introduction to network flows over time, *in* 'Research trends in combinatorial optimization', Springer, pp. 451–482.

Tyan, J. C., Wang, F.-K. and Du, T. C. (2003), An evaluation of freight consolidation policies in global third party logistics, *Omega* **31**(1), 55–62.

Ulku, M. A. (2009*a*), Analysis of shipment consolidation in the logistics supply chain, PhD thesis, University of Waterloo.

Ülkü, M. A. (2009*b*), Comparison of typical shipment consolidation programs: structural results, *Management Science and Engineering* **3**(4), 27–33.

Van Loon, J. (1981), Irreducibly inconsistent systems of linear inequalities, *European Journal of Operational Research* **8**(3), 283–288.

Vu, D. M., Hewitt, M., Boland, N. and Savelsbergh, M. (2020), Dynamic discretization discovery for solving the time-dependent traveling salesman problem with time windows, *Transportation science* **54**(3), 703–720.

Wang, X. and Regan, A. C. (2002), Local truckload pickup and delivery with hard time window constraints, *Transportation Research Part B: Methodological* **36**(2), 97–112.

Wang, X. and Regan, A. C. (2009), On the convergence of a new time window discretization method for the traveling salesman problem with time window constraints, *Computers & Industrial Engineering* **56**(1), 161–164.

Wieberneit, N. (2008), Service network design for freight transportation: a review, *OR spectrum* **30**(1), 77–112.

Proofs of statements and additional details about the computational experiments

EC.1. Proof of statements

EC.1.1. Proof of Proposition 1

The matrix associated with constraints (12b), (12c) and (12d) has coefficients in $\{0, 1, -1\}$ and each column of its transpose has exactly two nonzero elements of a different sign. Hence, it is totally unimodular and, together with the bound constraints (12e)-(12i), ensures that the extreme points of (12b)-(12i) are integral (see, for example Bertsimas and Weismantel 2005).□

EC.1.2. Proof of Theorem 1

Let $(\bar{x}, \bar{y}, \bar{w})$ be an optimal solution of formulation SND-HC($\mathcal{D}_{\hat{\mathcal{T}}}$) (i.e., an optimal CTSNDP-HC solution) of cost $\bar{z}$, and let $\overline{\mathcal{A}} = \{((i,t),(j,t+\tau_{ij})) \in \mathcal{A}_{\hat{\mathcal{T}}} : \overline{y}_{ij}^{t,t+\tau_{ij}} > 0\}$ be the set of arcs traversed by the commodities. Below we show that to solution $(\bar{x}, \bar{y}, \bar{w})$ corresponds a feasible, but not necessarily optimal, solution (x, y, w) of formulation SND-HC-R($\mathcal{D}_{\mathcal{T}}$) of cost $z = \bar{z}$.

By means of the mapping described by expressions (13), we can associate with vectors $\bar{x}$ and $\bar{y}$ and corresponding paths $\{\overline{P}^k\}_{k\in\mathcal{K}}$, solution vectors x and y. As shown by Boland et al. (2017), to solution vector $\bar{x}$ corresponds a set $\{P^k\}_{k\in\mathcal{K}}$ of feasible paths in network $\mathcal{D}_{\mathcal{T}}$, one path for each commodity, with the same total fixed and flow cost of paths $\{\overline{P}^k\}_{k\in\mathcal{K}}$. More precisely, for each commodity $k \in \mathcal{K}$ and path $\overline{P}^k = (a_1^k, \ldots, a_{\eta^k}^k)$, $a_h^k \in \overline{\mathcal{A}}$, $h = 1, \ldots, \eta^k$, with $a_h^k = ((i_h^k, t_h^k), (i_{h+1}^k, t_h^k + \tau_{i_h^k i_{h+1}^k}))$ and $t_{h+1}^k \geq t_h^k + \tau_{i_h^k i_{h+1}^k}$ for $h = 1, .., \eta^k - 1$ induced by solution vector $\bar{x}$, corresponds a feasible path $P^k = (\mu(a_1^k), \ldots, \mu(a_{\eta^k}^k))$ in $\mathcal{D}_{\mathcal{T}}$ with appropriate holding arcs. For each $k \in \mathcal{K}$ we have that the total transit time $T(P^k)$ of path P^k can be computed as

$$T(P^k) = T(\overline{P}^k) = \sum_{h=1}^{\eta^k} \tau_{a_h^k} = \sum_{((i,t),(j,\bar{t}))\in\overline{\mathcal{A}}} \tau_{ij}\overline{x}_{ij}^{kt\bar{t}} = \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}} \tau_{ij} x_{ij}^{kt\bar{t}}. \tag{EC.1}$$

The holding times $\bar{w}$ of solution $(\bar{x}, \bar{y}, \bar{w})$ can be computed as:

$$\overline{w}_i^k = \begin{cases} t_1^k - e^k, & i = o^k, \\ l^k - (t_{\eta^k}^k + \tau_{i_{\eta^k}^k d^k}), & i = d^k, \\ t_h^k - (t_{h-1}^k + \tau_{i_{h-1}^k i_h^k}), & i = i_h^k, h = 2, ..., \eta^k, \\ 0, & \text{otherwise}, \end{cases}, \forall\, i \in \mathcal{N}, \forall\, k \in \mathcal{K}.$$

It is easy to see that for each $k \in \mathcal{K}$ we have

$$\sum_{i\in\mathcal{N}} \overline{w}_i^k = l^k - e^k - T(\overline{P}^k). \tag{EC.2}$$

We now show that solution $w_i^k = \overline{w}_i^k$, $\forall i \in \mathcal{N}$, $k \in \mathcal{K}$, satisfies constraints (21)-(23), thus showing that to solution $(\bar{x}, \bar{y}, \bar{w})$ corresponds a feasible solution (x, y, w) of SND-HC-R($\mathcal{D}_{\mathcal{T}}$) of the same total fixed, flow and holding cost. For each path P^k, $k \in \mathcal{K}$, we have that equation (23) follows from expression (EC.1) and (EC.2). Constraints (21) and (22) for the values w_i^k and path P^k are as follows:

$$w_i^k \le \begin{cases} \xi^k(\mu(a_1^k)) - e^k, & i = o^k, \\ l^k - \psi^k(\mu(a_{\eta^k}^k)), & i = d^k, \\ \xi^k(\mu(a_h^k)) - \psi^k(\mu(a_{h-1}^k)), & i = i_h^k, h = 2, ..., \eta^k, \\ 0, & \text{otherwise}, \end{cases} \quad \forall\, i \in \mathcal{N}, \tag{EC.3}$$

$$w_i^k \ge \begin{cases} \vartheta^k(\mu(a_1^k)) - e^k, & i = o^k, \\ l^k - \delta^k(\mu(a_{\eta^k}^k)), & i = d^k, \\ \vartheta^k(\mu(a_h^k)) - \delta^k(\mu(a_{h-1}^k)), & i = i_h^k, h = 2, ..., \eta^k, \\ 0, & \text{otherwise}, \end{cases} \quad \forall\, i \in \mathcal{N}, \tag{EC.4}$$

where for each a_h^k, $h = 1, ..., \eta^k$,

$$\vartheta^k(\mu(a_h^k)) = \max\{\rho_{i_h^k}(t_h^k), e^k + \phi^k(o^k, i_h^k)\}, \tag{EC.5}$$

$$\xi^k(\mu(a_h^k)) = \begin{cases} \min\{\vec{t}_{i_h^k}(\rho_{i_h^k}(t_h^k)), l^k - \tau_{i_h^k i_{h+1}^k} - \phi^k(i_{h+1}^k, d^k)\}, & \text{if } \vec{t}_{i_h^k}(\rho_{i_h^k}(t_h^k)) - \rho_{i_h^k}(t_h^k) > 1, \\ \min\{\rho_{i_h^k}(t_h^k), l^k - \tau_{i_h^k i_{h+1}^k} - \phi^k(i_{h+1}^k, d^k)\}, & \text{otherwise}, \end{cases} \tag{EC.6}$$

$$\psi^k(\mu(a_h^k)) = \max\{\rho_{i_h^k}(t_h^k) + \tau_{i_h^k i_{h+1}^k}, e^k + \phi^k(o^k, i_h^k) + \tau_{i_h^k i_{h+1}^k}\}, \tag{EC.7}$$

$$\delta^k(\mu(a_h^k)) = \xi^k(\mu(a_h^k)) + \tau_{i_h^k i_{h+1}^k}. \tag{EC.8}$$

Based on the mapping functions $\rho(.)$, $\sigma(.)$ and with the fact that $e^k + \phi^k(o^k, i_h^k) \le t_h^k \le l^k - \tau_{i_h^k i_{h+1}^k} - \phi^k(i_{h+1}^k, d^k)$, $h = 1, ..., \eta^k$, we have that:

(i) for each a_h^k, $h = 1, ..., \eta^k - 1$, $\vartheta^k(\mu(a_h^k)) \le t_h^k$ and $\xi^k(\mu(a_h^k)) \ge t_h^k$;

(ii) for each a_h^k, $h = 2, ..., \eta^k$, $\psi^k(\mu(a_h^k)) \le t_h^k + \tau_{i_h^k i_{h+1}^k}$ and $\delta^k(\mu(a_h^k)) \ge t_h^k + \tau_{i_h^k i_{h+1}^k}$.

Thus, we can show that:

(i) $\xi^k(\mu(a_1^k)) - e^k \ge t_1^k - e^k \ge \vartheta^k(\mu(a_1^k)) - e^k$;

(ii) $l^k - \psi^k(\mu(a_{\eta^k}^k)) \ge l^k - (t_{\eta^k}^k + \tau_{i_{\eta^k}^k d^k}) \ge l^k - \delta^k(\mu(a_{\eta^k}^k))$;

(iii) $\xi^k(\mu(a_h^k)) - \psi^k(\mu(a_{h-1}^k)) \ge t_h^k - (t_{h-1}^k + \tau_{i_{h-1}^k i_h^k}) \ge \vartheta^k(\mu(a_h^k)) - \delta^k(\mu(a_{h-1}^k))$, $h = 2, ..., \eta^k$;

(iv) For each $i \notin P^k$, we have $w_i^k \le 0$ for constraint (EC.3) and $w_i^k \ge 0$ for constraint (EC.4), hence $w_i^k = 0$.

Solution (x, y, w) is then proved to be a feasible SND-HC-R($\mathcal{D}_\mathcal{T}$) solution of the same cost as solution $(\bar{x}, \bar{y}, \bar{w})$.

EC.1.3. Proof of Theorem 2

Let $(\overline{x}, \overline{y}, \overline{w})$ be a feasible solution of formulation SND-HC-R($\overline{\mathcal{D}}_\mathcal{T}$) of cost $\overline{z}$. We show that to solution $(\overline{x}, \overline{y}, \overline{w})$ corresponds a feasible, but not necessarily optimal, solution (x, y, w) of SND-HC-R($\mathcal{D}_\mathcal{T}$), of cost z such that $\overline{z} = z$.

Let $\overline{\mathcal{A}} = \{((i,t),(j,t')) \in \overline{\mathcal{A}}_\mathcal{T} : \overline{y}_{ij}^{tt'} > 0\}$ be the set of arcs traversed by solution $(\overline{x}, \overline{y}, \overline{w})$.

Consider an arc $((i,t),(j,t')) \in \overline{\mathcal{A}}$ such that all arcs of the form $((i,t),(j,t''))$ belong to $\mathcal{A}_{\mathcal{T}}$ with $t'' < t'$. If no such arc $((i,t),(j,t'))$ exists, solution vectors $x = \overline{x}$ and $y = \overline{y}$ are clearly feasible for constraints (5), (6), (8), and (9). If such arc $((i,t),(j,t'))$ exists, since networks $\overline{\mathcal{D}}_{\mathcal{T}}$ and $\mathcal{D}_{\mathcal{T}}$ are defined on the same set of time points $\mathcal{N}_{\mathcal{T}}$, a path from (j,t'') and (j,t') exists in $\mathcal{D}_{\mathcal{T}}$.

Initialize $x = 0$ and $y = 0$ and define $x_{ij}^{kt\bar{t}} = \overline{x}_{ij}^{kt\bar{t}}$, $k \in \mathcal{K}$, and $y_{ij}^{t\bar{t}} = \overline{y}_{ij}^{t\bar{t}}$ for all arcs in $((i,t),(j,\bar{t})) \in (\mathcal{H}_{\mathcal{T}} \cup \mathcal{A}_{\mathcal{T}}) \cap (\overline{\mathcal{H}}_{\mathcal{T}} \cup \overline{\mathcal{A}}_{\mathcal{T}})$. We can adapt the solution $(\overline{x}, \overline{y}, \overline{w})$ with regard to the arc $((i,t),(j,t'))$ to the solution (x,y,w) concerning the arc $((i,t),(j,t''))$, with the addition of the holding arcs joining (j,t'') to (j,t')), by setting $y_{ij}^{tt''} = \overline{y}_{ij}^{tt'}$ and $x_{ij}^{ktt''} = x_{jj}^{kt''t'} = \overline{x}_{ij}^{ktt'}$. The resulting (x,y,w) solution is also feasible for constraints (5), (6), (8), and (9), and the process can be repeated for every arc $((i,t),(j,t')) \in \overline{\mathcal{A}}_{\mathcal{T}}$ with $t'' < t'$ for all $((i,t),(j,t'')) \in \mathcal{A}_{\mathcal{T}}$. For each commodity $k \in \mathcal{K}$, let $\overline{P}^k = (\overline{a}_1^k, \ldots, \overline{a}_{\eta^k}^k)$, $\overline{a}_h^k \in \overline{\mathcal{A}}$, $h = 1, \ldots, \eta^k$, be the path induced by solution $(\bar{x}, \bar{y})$ with $\overline{a}_h^k = ((i_h^k, \bar{t}_h^k), (i_{h+1}^k, \overline{\pi}_{h+1}^k))$, $h = 1, \ldots, \eta^k$, where $\bar{t}_h^k$ is the departure time at node i_h^k and $\overline{\pi}_h^k$ is the corresponding arrival time. Due to the definition of solution vectors (x,y) based on solution vectors $(\overline{x}, \overline{y})$, in graph $\mathcal{D}_{\mathcal{T}}$ for commodity k we have a path $P^k = (a_1^k, \ldots, a_{\eta^k}^k)$, $a_h^k \in \mathcal{A}_{\mathcal{T}}$, with departure time $t_h^k = \bar{t}_h^k$, $h = 1, \ldots, \eta^k$, and arrival times $\pi_h^k \le \overline{\pi}_h^k$, $h = 2, \ldots, \eta^k + 1$.

We now show that solution vector $w = \overline{w}$ satisfies constraints (21)-(23) of formulation SND-HC-R($\mathcal{D}_{\mathcal{T}}$), thus showing that (x,y,w) is a feasible SND-HC-R($\mathcal{D}_{\mathcal{T}}$) solution having the same cost of solution $(\overline{x}, \overline{y}, \overline{w})$. First, for each $k \in \mathcal{K}$, $T(\overline{P}^k) = T(P^k)$, hence equations (23) are satisfied. Then, we have that solution vector $\overline{w}$ is defined as:

$$\overline{w}_i^k \le \begin{cases} \xi^k(\overline{a}_1^k) - e^k, & i = o^k, \\ l^k - \psi^k(\overline{a}_{\eta^k}^k), & i = d^k, \\ \xi^k(\overline{a}_h^k) - \psi^k(\overline{a}_{h-1}^k), & i = i_h^k, h = 2, ..., \eta^k, \\ 0, & \text{otherwise}, \end{cases} \quad \forall\, i \in \mathcal{N}, \tag{EC.9}$$

$$\overline{w}_i^k \ge \begin{cases} \vartheta^k(\overline{a}_1^k) - e^k, & i = o^k, \\ l^k - \delta^k(\overline{a}_{\eta^k}^k), & i = d^k, \\ \vartheta^k(\overline{a}_h^k) - \delta^k(\overline{a}_{h-1}^k), & i = i_h^k, h = 2, ..., \eta^k, \\ 0, & \text{otherwise}, \end{cases} \quad \forall\, i \in \mathcal{N}, \tag{EC.10}$$

where for each $\overline{a}_h^k$, $h = 1, ..., \eta^k$,

$$\vartheta^k(\overline{a}_h^k) = \max\{\bar{t}_h^k, e^k + \phi^k(o^k, i_h^k)\}, \tag{EC.11}$$

$$\xi^k(\overline{a}_h^k) = \begin{cases} \min\{\vec{t}_{i_h^k}(\bar{t}_h^k), l^k - \tau_{i_h^k i_{h+1}^k} - \phi^k(i_{h+1}^k, d^k)\}, & \text{if } \vec{t}_{i_h^k}(\bar{t}_h^k) - \bar{t}_h^k > 1, \\ \min\{\bar{t}_h^k, l^k - \tau_{i_h^k i_{h+1}^k} - \phi^k(i_{h+1}^k, d^k)\}, & \text{otherwise}, \end{cases} \tag{EC.12}$$

$$\psi^k(\overline{a}_h^k) = \max\{\bar{t}_h^k + \tau_{i_h^k i_{h+1}^k}, e^k + \phi^k(o^k, i_h^k) + \tau_{i_h^k i_{h+1}^k}\}, \tag{EC.13}$$

$$\delta^k(\overline{a}_h^k) = \xi^k(\overline{a}_h^k) + \tau_{i_h^k i_{h+1}^k}, \tag{EC.14}$$

and for inequalities (21) and (22) we have

$$w_i^k \leq \begin{cases} \xi^k(a_1^k) - e^k, & i = o^k, \\ l^k - \psi^k(a_{\eta^k}^k), & i = d^k, \\ \xi^k(a_h^k) - \psi^k(a_{h-1}^k), & i = i_h^k, h = 2, ..., \eta^k, \\ 0, & \text{otherwise}, \end{cases} \quad \forall\, i \in \mathcal{N}, \tag{EC.15}$$

$$w_i^k \geq \begin{cases} \vartheta^k(a_1^k) - e^k, & i = o^k, \\ l^k - \delta^k(a_{\eta^k}^k), & i = d^k, \\ \vartheta^k(a_h^k) - \delta^k(a_{h-1}^k), & i = i_h^k, h = 2, ..., \eta^k, \\ 0, & \text{otherwise}, \end{cases} \quad \forall\, i \in \mathcal{N}, \tag{EC.16}$$

where for each a_h^k, $h = 1, ..., \eta^k$,

$$\vartheta^k(a_h^k) = \max\{t_h^k, e^k + \phi^k(o^k, i_h^k)\}, \tag{EC.17}$$

$$\xi^k(a_h^k) = \begin{cases} \min\{\vec{t}_{i_h^k}(t_h^k), l^k - \tau_{i_h^k i_{h+1}^k} - \phi^k(i_{h+1}^k, d^k)\}, & \text{if } \vec{t}_{i_h^k}(t_h^k) - t_h^k > 1, \\ \min\{t_h^k, l^k - \tau_{i_h^k i_{h+1}^k} - \phi^k(i_{h+1}^k, d^k)\}, & \text{otherwise}, \end{cases} \tag{EC.18}$$

$$\psi^k(a_h^k) = \max\{t_h^k + \tau_{i_h^k i_{h+1}^k}, e^k + \phi^k(o^k, i_h^k) + \tau_{i_h^k i_{h+1}^k}\}, \tag{EC.19}$$

$$\delta^k(a_h^k) = \xi^k(a_h^k) + \tau_{i_h^k i_{h+1}^k}, \tag{EC.20}$$

since networks $\overline{\mathcal{D}}_{\mathcal{T}}$ and $\mathcal{D}_{\mathcal{T}}$ are defined on the same set of time points $\mathcal{N}_{\mathcal{T}}$ and $t_h^k = \bar{t}_h^k, h = 1, \ldots, \eta^k$, we have that $\vartheta^k(a_h^k) = \vartheta^k(\bar{a}_h^k)$, $\xi^k(a_h^k) = \xi^k(\bar{a}_h^k)$, $\psi^k(a_h^k) = \psi^k(\bar{a}_h^k)$ and $\delta^k(a_h^k) = \delta^k(\bar{a}_h^k)$ for all $h = 1, \ldots, \eta^k$. Constraints (EC.15) and (EC.16) thus have the same right-hand side as constraints (EC.9) and (EC.10), respectively. Hence, $w = \overline{w}$ is proved to be a feasible solution for inequalities (21) and (22).□

EC.1.4. Proof of Lemma 1

Based on the optimality condition (i), we have that $\psi^k(a) = \bar{t}$ for all $a = ((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}$ with $x_{ij}^{kt\bar{t}} = 1$ for some $k \in \mathcal{K}$. The flat solution $\mathcal{S}$ associated with solution (x, y, w) must be implementable because t and $\bar{t}$ associated with $x_{ij}^{kt\bar{t}} = 1$ are feasible arrival and departure times to commodity k in solution $\mathcal{S}$. Each θ_i^k is therefore the corresponding holding time for commodity k at terminal i. This implies that (x, y, w) with $w_i^k = \theta_i^k$ for all $i \in \mathcal{N}$, $k \in \mathcal{K}$, is a feasible solution to the CTSNDP-HC. Since (x, y, w) is an optimal solution of relaxation SND-HC-R($\mathcal{D}_{\mathcal{T}}$), (x, y, w) is therefore proved to be an optimal CTSNDP-HC solution of cost LB. Note that Algorithm 2 computes $UB = \hat{z}_{fc} + \hat{z}_w$ and we have that $LB = z_{fc}(\mathcal{S}) + z_w(LB)$. Because the flat solution $\mathcal{S}$ associated with solution (x, y, w) is implementable, we have that $\hat{z}_{fc} = z_{fc}(\mathcal{S})$. Moreover, since Algorithm 2 solves model $IM(\mathcal{S})$ minimizing the total holding cost associated with the flat solution, we also have that $\hat{z}_w = z_w(LB)$, hence $LB = UB$.□

EC.2. Additional details about the computational experiments

EC.2.1. Implementation details

It is worth noting that DDD-based methods rely on two (relative) optimality tolerances: (i) the optimality tolerance used by the DDD algorithm (see parameter *optimality tolerance* of the exact algorithm described in §4), and (ii) the optimality tolerance used by the MIP solver. For the sake of the notation, we denote by tol_{DDD} and tol_{MIP} the two optimality tolerances, respectively. For all considered instances in the two sets of experiments in Section 5, we used $tol_{DDD} = 0.01$ for both EXM and EXM-0, $tol_{MIP} = 0.01$ for EXM-0. For algorithm EXM, based on our preliminary experiments and as reported by Marshall et al. (2021), we also found to be computationally convenient to dynamically change parameter tol_{MIP}, that starts with 0.04 and is computed as $\max\{gap \times 0.25, 0.098\}$ for each later iteration, where gap is the final gap of the previous iteration. Also, in these experiments, a time limit of two hours is conducted for both EXM and EXM-0. The time limit is imposed over all the iterations, meaning that at each iteration the time limit imposed on the MIP solver is the remaining time. Given tolerance tol_{MIP} applied to the MIP solver and in order to compute safe lower bounds, the lower bound value LB is set equal to the best known bound on the optimal objective given by the Gurobi solver at termination through parameter `ObjBound`.

The algorithms EXM and EXM-0 were implemented in Java language, and Gurobi (v.8.1.1) (Gurobi Optimization 2021) was used as the LP solver to solve model $IM(\mathcal{S})$, and as the MIP solver to solve relaxation SND-HC-R($\mathcal{D}_{\mathcal{T}}$). The Gurobi function `Model.computeIIS()` was used to compute IISs in Algorithm 2. The experiments were performed on an Intel(R) Core(TM) i7-8700 (3.20 GHz) Desktop PC equipped with 64 GB RAM running under Windows 10 64-bit operating system.

EC.2.2. Generating instances based on CTSNDP benchmark instances

Table EC.1 gives the details of the 31 classes of networks $\mathcal{D} = (\mathcal{N}, \mathcal{A})$ considered by Crainic et al. (2001). In the table, the column "Cost ratio" computed as $\frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} \frac{f_a}{c_a u_a}$ measures the ratio between fixed and variable costs, "Cap ratio" computed as $\sum_{k\in\mathcal{K}} q^k / \frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} u_a$ indicates whether the arcs are loosely or tightly capacitated, and "Avg length" computed as $\frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}} \underline{\tau}_{o^k d^k}$, where $\underline{\tau}_{o^k d^k}$ is the length of the least total travel time path from o^k to d^k, is the average length of the least total travel time paths. For each of the 31 classes of networks reported in the table, Boland et al. (2018) generated 18 CTSNDP timed instances by first calculating the travel times for each arc and then by generating the time windows for each commodity by randomly sampling from a normal distribution.

For each of the 558 CTSNDP instances, we generated a CTSNDP-HC instance by setting the per-unit-of-demand-and-time holding cost h_i^k for each commodity $k \in \mathcal{K}$ at each terminal $i \in \mathcal{N}$ as follows. We set the holding cost using the scheme proposed by Lai et al. (2022), in which fixed,

Table EC.1 Instances from Crainic et al. (2001)

Class	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{K}\rvert$	Cost ratio	Cap ratio	Avg length
c33	20	228	39	0.02	5.8	2407.9
c35	20	230	40	0.02	16.0	767.9
c36	20	230	40	0.08	16.0	3705.8
c37	20	228	200	0.51	16.0	1871.4
c38	20	230	200	0.97	16.0	4381.0
c39	20	229	200	0.47	20.0	1691.3
c40	20	228	200	0.94	22.0	3522.1
c41	20	288	40	0.02	8.0	1622.0
c42	20	294	40	0.08	10.0	5675.8
c43	20	294	40	0.02	16.0	776.5
c44	20	294	40	0.08	16.0	3517.9
c45	20	294	200	0.48	25.0	1124.2
c46	20	292	200	1.01	25.0	2632.0
c47	20	291	200	0.46	28.0	996.6
c48	20	291	200	0.95	28.0	2271.6
c49	30	518	100	0.10	20.0	341.1
c50	30	516	100	0.51	20.0	1586.5
c51	30	519	100	0.09	29.9	206.6
c52	30	517	100	0.49	29.9	1161.5
c53	30	520	400	0.18	40.0	612.1
c54	30	520	400	0.36	40.0	1061.8
c55	30	516	400	0.18	49.9	479.4
c56	30	518	400	0.35	49.9	966.9
c57	30	680	100	0.09	20.0	307.6
c58	30	680	100	0.20	20.0	592.8
c59	30	687	100	0.10	29.9	187.1
c60	30	686	100	0.20	29.9	394.7
c61	30	685	400	0.19	40.0	503.8
c62	30	679	400	0.36	40.0	1056.5
c63	30	678	400	0.18	49.9	381.4
c64	30	683	400	0.34	49.9	780.0

flow, and holding costs were defined for the LTL shipment case. We observe that the average per-unit-of-demand-and-time transportation cost for the 558 CTSNDP instances, calculated as $\frac{\sum_{a\in\mathcal{A}}((c_a+f_a/u_a)/\tau_a)}{|\mathcal{A}|}$, is nearly 0.02. The value obtained is comparable with the transportation cost per hundred weights per unit of Euclidean distance, nearly 0.03, used in Lai et al. (2022). Note that Lai et al. (2022) utilized a discount cost function for the flow cost expressed in dollars per hundred weights. The transportation cost per hundred weight per unit of Euclidean distance can be approximated as $\frac{\sum_{a\in\mathcal{A}}((c_a^{max}+(100\,f_a)/u_a)/d_a)}{|\mathcal{A}|}$, where c_a^{max} represents the maximum unit charge and d_a represents the Euclidean distance of the arc. In Lai et al. (2022), the holding cost per weight per time period is randomly sampled in the interval [0.025, 0.1], with a holding cost per hundred weights per unit of Euclidean distance ranging from 0.0025 to 0.02. In addition, the arc travel times are computed as a function of the Euclidean distances associated with the arcs. Therefore, we set the per-unit-of-demand-and-time holding cost as follows. For each commodity $k\in\mathcal{K}$, we randomly sample value h^k from the interval [0.0025, 0.02], then for each terminal $i\in\mathcal{N}$, the per-unit-of-demand-and-time holding cost h_i^k is randomly sampled from $[0.8\times h^k, 1.2\times h^k]$. Since, in practice, a commodity does not incur any holding costs after it reaches the destination, for each commodity $k \in \mathcal{K}$ we set $h^k_{d^k} = 0$. We, therefore, generated 558 CTSNDP-HC instances.

EC.2.3. Newly generated CTSNDP-HC instances

The procedure used to generate the new instances follows two main steps.

(1) **Varying the connectivity level**. For each instance of Table EC.1 and the corresponding network $\mathcal{D} = (\mathcal{N}, \mathcal{A})$, we first derive a timed instance using the method proposed by Boland et al. (2018), i.e., we compute the travel times τ_{ij}. We then generate the per-unit-of-demand-and-time cost h^k_i using the method explained in §EC.2.2. Let Γ be the time of the path in $\mathcal{D}$ having maximum time among the least time $o^k - d^k$ paths associated with the vertices o^k, d^k, $\forall k \in \mathcal{K}$. Then, we reduce the number of arcs of the network $\mathcal{D} = (\mathcal{N}, \mathcal{A})$ by an arc reduction procedure that at each iteration performs the following steps:

   (i) Randomly select an arc a by means of a uniform distribution from the set of arcs $\mathcal{A}$.

   (ii) Check the connectivity of the network $(\mathcal{N}, \mathcal{A} \setminus \{a\})$, i.e., check if for every pair of vertices o^k, d^k, $k \in \mathcal{K}$, there exists a path connecting o^k to d^k.

   (iii) If the graph is connected, set $\mathcal{A} = \mathcal{A} \setminus \{a\}$ and begin a new iteration.

   If the removal of an arc a results in a disconnected network, a new arc is randomly selected and the procedure terminates after $\frac{1}{10}x$ unsuccessful removal attempts, where x is the initial number of arcs. After termination, each remaining arc in the resulting graph $\mathcal{D}$ is in turn selected and tested for removal.

   Let NR be the total number of arcs removed. We consider four final networks corresponding to the networks obtained after the removal of $\lfloor xNR \rfloor$ arcs where $x \in \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$, denoted as $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$ and $\mathcal{D}_4$, respectively.

(2) **Varying the flexibility level**. Given a network $\mathcal{D}_x$, $x = 1, 2, 3, 4$, let $\underline{\tau}_{ij}$, $i, j \in \mathcal{N}$, $i \neq j$ be the length of the least total travel time path from i to j, and let $\mathcal{B} = \{(i, j) : \underline{\tau}_{ij} \leq \Gamma\}$.

   If, for a commodity $k \in \mathcal{K}$, we have $\underline{\tau}_{o^k d^k} > \Gamma$, we assign to the commodity new origin and destination nodes by randomly sampling with a uniform distribution a new pair from set $\mathcal{B}$, and we recompute the new value $\underline{\tau}_{o^k d^k}$.

   We then generate earliest and latest times also based on the method proposed by Boland et al. (2018) as follows:

   (a) We compute the average length computed as $l_{avg} = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \underline{\tau}_{o^k d^k}$.

   (b) For generating values e^k, we create a normal distribution with mean l_{avg} and standard deviation $\frac{1}{6} l_{avg}$.

   (c) For generating values l^k, we create three normal distributions (denoted as A, B and C, respectively) from which we drawn values l^k, all of which are defined by a standard deviation $\frac{1}{6}\mu$ but where we consider the values for the mean μ, $\frac{1}{2}l_{avg}$, l_{avg} and $\frac{3}{2}l_{avg}$. A value l^k is set equal to $e^k + \tau_{o^k d^k} + \alpha$ where α is the value drawn from a distribution.

Based on the above two steps, for each of the instances in Table EC.1, we generate four different networks $\mathcal{D}_x$, $x = 1,2,3,4$, and three instances based on the three different distributions for values e^k and l^k. These steps are repeated three times to finally obtain a total of $3 \times (31 \times 4 \times 3) = 1116$ instances.

EC.2.4. Preliminary results for EXM-0

Here we summarize the results obtained by the baseline algorithm EXM-0 in solving the 558 CTSNDP instances, and we then compare its performance with the results of BHMS17 and MBSH21 reported in Marshall et al. (2021). A time limit of one hour was imposed to EXM-0, as done for both BHMS17 and MBSH21.

In the comparison reported by Marshall et al. (2021), for method BHMS17, tol_{DDD} and tol_{MIP} were set equal to 0.01 and 0.0001 (i.e., the CPLEX default setting), respectively. For method MBSH21, tol_{DDD} was also set equal to 0.01, whereas tolerance tol_{MIP} was dynamically changed during the different iterations. More precisely, the initial tolerance is set equal to 0.04, and then for each iteration tol_{MIP} is computed as $\max\{gap \times 0.25, tol_{DDD} \times 0.98\}$, where gap is the final gap of the previous iteration. Note that EXM-0 applies the same adaptive optimality tolerance. For the set of CTSNDP instances, our comparison is based on the results reported by Marshall et al. (2021) which were obtained on a single core machine using CPLEX 12.6 as the MIP solver (for both BHMS17 and MBSH21, with no specific machine type being reported). Because the computational environment of BHMS17 and MBSH21 was different from that of our algorithms, a direct comparison is therefore not possible. However, in what follows we give a clear overall picture of the relative performance, especially when the total number of instances solved to proven optimality is compared.

Table EC.2 gives the comparison of the three algorithms. For each group of instances, the table shows the number of instances in the group and, for each algorithm, the average percentage deviation of the final upper bound UB computed with respect to the final lower bound LB ("$\%UB$"), i.e., $100.0 \times \frac{UB-LB}{UB}$, the average computing time in seconds ("*time*"), the average number of iterations ("*iter*"), and the percentage of the instances solved to optimality ("%*opt*") (within the given optimality tolerance). The average values over all instances are computed. For each method, the upper bound UB corresponds to the cost of the best solution found by the method over the different algorithm iterations and the lower bound LB is computed as the maximum among the lower bounds computed at the different iterations.

The table shows that our implementation of the algorithm of Boland et al. (2017) compares well with both algorithms BHMS17 and MBSH21, and also shows similar performance on the different

Table EC.2 Summary results on the CTSNDP instances

Group	Algorithm	$\%UB$	*time*	*iter*	$\%opt$
HC/LF	BHMS17	0.08	1391.1	5.3	77.1
183	MBSH21	0.12	677.8	14.8	85.8
	EXM-0	1.00	236.3	10.0	96.7
HC/HF	BHMS17	0.56	1966.7	6.0	53.7
177	MBSH21	0.84	1693.8	17.5	56.5
	EXM-0	3.18	1555.5	11.0	65.0
LC/LF	BHMS17	0.00	28.6	3.7	100.0
94	MBSH21	0.00	0.6	6.5	100.0
	EXM-0	0.70	0.5	3.5	100.0
LC/HF	BHMS17	0.00	1.5	2.5	100.0
104	MBSH21	0.00	0.1	3.2	100.0
	EXM-0	0.55	0.1	1.6	100.0

groups of instances. We note that the computational environment of EXM-0 is different from the one used by the other methods. However, the comparison over the total number of instances solved to proven optimality gives a clear global picture of the relative performance.

In groups **LC/LF** and **LC/HF**, EXM-0 was capable of solving to optimality all the instances within the imposed optimality tolerance. In these groups, with respect to BHMS17 and MBSH21, EXM-0 shows higher percentage deviations of the final upper bound UB. This can be due to the fact that different MIP solvers are used, and that EXM-0 computes a safe lower bound based on the best-known bound given by the Gurobi MIP solver. Based on the results of Table EC.2, we adopted algorithm EXM-0 as a baseline algorithm for comparison purposes with EXM.

EC.2.5. Additional results on the effect of the algorithm components

The heuristic method described by Algorithm 2 removes at each iteration the infeasible consolidation constraint (12d) having the minimum increase of the fixed cost. To illustrate the effectiveness of our choice, below, we compare alternative methods to handle infeasible consolidation constraints (12d). More specifically, we consider alternative upper bounds, namely $UBM1$ and $UBM2$, where the infeasible constraint during Algorithm 2 is removed based on the following rules: (i) $UBM1$: the infeasible constraint with the earliest departure time. (ii) $UBM2$: the infeasible constraint with the tightest time flexibility on the selected consolidation arc (i,j), i.e., $(l^k - \phi^k(j,d^k) - \tau_{ij}) - (e^k + \phi^k(o^k,i))$. In addition, we also consider upper bound $UB1$ described in Section 5.1.2. Table EC.3 gives the comparison, where $\%UBx = 100.0 \times \frac{UBx - UB}{UBx}$, $x \in \{M1, M2, 1\}$, is the percentage deviations of upper bound UBx with respect to the upper bound UB computed by Algorithm 2. The comparison concerns the different upper bounds $UBM1$, $UBM2$, and UB computed at the first iteration of algorithm EXM and upper bound $UB1$ computed at the first iteration of algorithm EXM-0. The table shows that the different upper bounds show similar results in the LC instances. In contrast, the upper bound UB is significantly better than the other upper bounds in the HC instances, thus attesting to the effectiveness of our choice.

Table EC.3 Computational results for EXM variants with different refinement strategies

	$\%UBM1$	$\%UBM2\$$	$\%UB1$
HC/LF	0.9	1.1	0.9
HC/HF	2.5	2.9	2.0
LC/LF	0.1	0.1	0.1
LC/HF	0.1	0.1	0.1

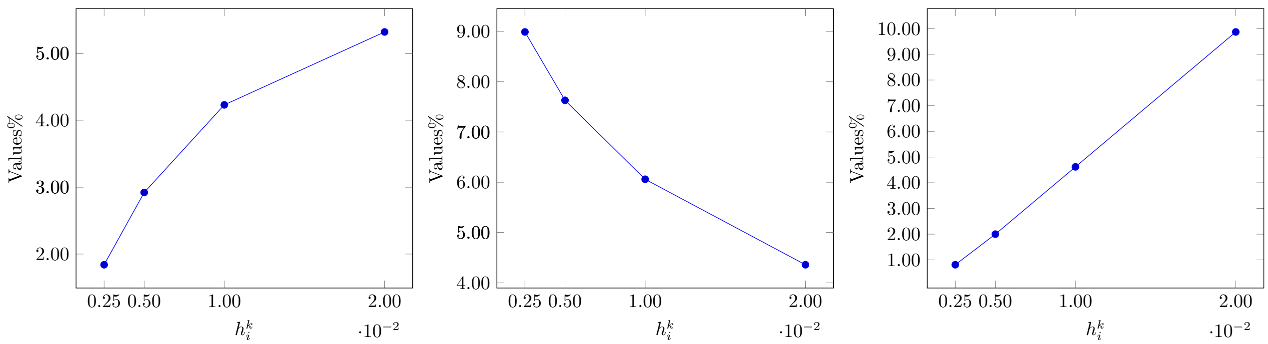
To attest the effectiveness of the refinement strategies, we executed algorithm EXM with refining different sets of short-arcs. More preciously, we conducted algorithm EXM without refining the short-arcs identified by Algorithm 2 in refinement strategy 1, referred to as EXM_R1 and by selecting at most $|\mathcal{K}|/10$ short-arcs for the additional set in refinement strategy 1, referred to as EXM_R2. In the case of EXM_R2, we select fewer arcs than the limit of $|\mathcal{K}|/5$ arcs used by EXM. The results are shown in Table EC.4 using the same notation introduced in Table 2. The results show that the algorithm EXM benefits from a slightly aggressive refinement strategy 1 that refines all short-arcs identified by Algorithm 2 and refines more short-arcs for the additional set. Indeed, EXM shows a better optimality rate and optimality gap for unsolved instances compared to EXM_R1 and EXM_R2.

Table EC.4 Computational results for EXM variants with different refinement strategies

Group	Algorithm	$\%UB$			*time*	*iter*	*%opt*
		min	*max*	*avg*			
HC/LF	EXM_R1	1.17	1.9	1.7	277.2	4.4	98.4
	EXM_R2	1.10	2.0	1.6	372.2	6.1	97.8
	EXM	1.08	1.9	1.6	279.3	4.4	98.4
HC/HF	EXM_R1	1.01	8.9	3.4	3107.7	8.7	60.5
	EXM_R2	1.00	9.9	3.4	3118.0	8.9	62.7
	EXM	1.01	6.1	2.9	2902.7	6.2	65.5
LC/LF	EXM_R1	-	-	-	0.7	1.8	100.0
	EXM_R2	-	-	-	0.6	1.7	100.0
	EXM	-	-	-	0.7	1.8	100.0
LC/HF	EXM_R1	-	-	-	0.1	2.3	100.0
	EXM_R2	-	-	-	0.2	3.3	100.0
	EXM	-	-	-	0.2	2.3	100.0

EC.2.6. Sensitive analysis on the holding costs

To analyze the effect of varying the per-unit-of-demand-and-time cost h_i^k on the CTSNDP-HC we compare the results obtained by EXM for the case with different per-unit-of-demand-and-time cost h_i^k, i.e., $h_i^k \in \{0.0025, 0.05, 0.01, 0.02\}$. For the experiments, we considered a restricted set of instances composed of the instances of networks $\mathcal{D}_2$ and $\mathcal{D}_3$ under the three different distributions A, B, and C.



(a) $\%hc$ (b) $\%ht$ (c) $\%UB1_{avg}$

Figure EC.1 Sensitive analysis on the per-unit-of-demand-and-time cost h_i^k

Figure EC.1 summarizes the results obtained. For each per-unit-of-demand-and-time cost h_i^k (x axis), the figure shows the following average percentage values: (i) $\%hc$, the holding cost over the total solution cost, (ii) $\%ht$, the holding time over the total transit time, and (iii) $\%UB1_{avg}$, the average percentage deviation of upper bound $UB1$ with respect to upper bound UB. The percentage values were computed over all instances solved to optimality with all the considered unit holding values. The figure shows that when unit holding cost increases, both $\%UB1_{avg}$ and $\%hc$ increase. This implies that although a higher per-unit-of-demand-and-time holding cost induces a larger percentage of the holding cost, it induces a more significant cost saving achieved by UB over $UB1$. From the figure, we can also observe that when unit holding cost increases, $\%ht$ decreases. This implies that a higher per-unit-of-demand-and-time holding cost also induces a significant reduction in holding time. Hence, as per-unit-of-demand-and-time holding cost increases, it is more beneficial to take into account holding costs for solving the CTSNDP-HC. We notice that, with a per-unit-of-demand-and-time holding cost of 0.0025, the percentage deviation of upper bound $UB1$ is less than 1%, thus on average, a per-unit-of-demand-and-time holding cost around or below 0.0025 will make a negligible difference between UB and $UB1$ on the considered instances. However, the final holding cost also depends on other cost ratios, time flexibility, and network connectivity. Thus, the exact cut-off value for per-unit-of-demand-and-time holding costs differs for different instances.