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
Department of Electrical, Electronic, and Information Engineering "Guglielmo Marconi", University of Bologna, Italy, r.baldacci@unibo.it

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 studies on 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: Updated on September 30, 2021

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)

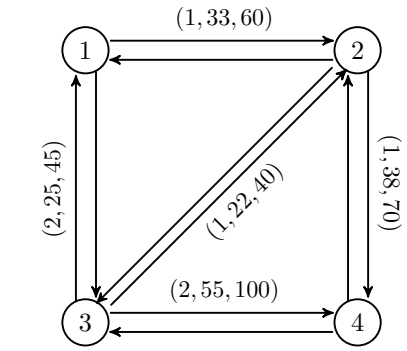
a per-unit-of-flow cost $c_{ij}^k \in \mathbb{R}_{>0}$ for each commodity $k \in \mathcal{K}$; (iii) a fixed cost $f_{ij} \in \mathbb{R}_{>0}$; and (iv) a capacity $u_{ij} \in \mathbb{N}_{>0}$. Installing one link on arc (i, j) provides a capacity u_{ij} at a cost f_{ij}. With each commodity $k \in \mathcal{K}$ is also associated an earliest available time $e^k \in \mathbb{N}_{>0}$ at the origin and a latest arrival time $l^k \in \mathbb{N}_{>0}$ 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

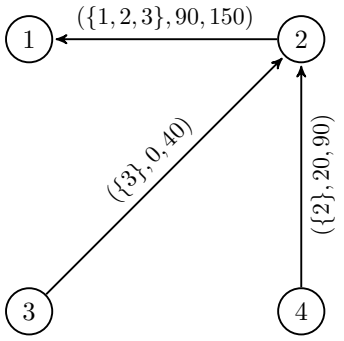
For many practical applications of the SNDP, *holding costs* have a significant impact on the service and consolidation decisions (Tyan et al. 2003, Bookbinder and Higginson 2002, Rudi et al. 2016, Hu et al. 2018). These costs are also called *consolidation penalty costs* (Ülkü 2009*b*), and can be facility-specific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



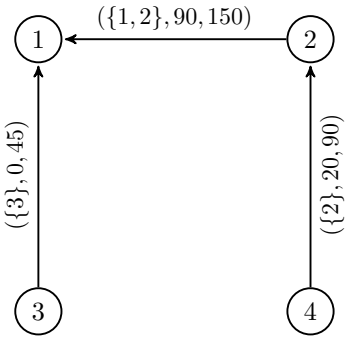
(a) Network $\mathcal{D}$: (c, f, τ) on arcs

k	o^k	d^k	q^k	e^k	l^k
1	2	1	25	0	160
2	4	1	30	20	180
3	3	1	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)

are generally modeled by properly defining the costs associated with the service and holding arcs of the network.

To highlight the importance of considering the holding costs in the SNDP, Figure 1 gives a simple example of a SNDP instance with a consideration of holding costs. The example involves three commodities, and 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, and the per-unit-of-flow cost of arc (i, j) is the same for all three commodities. 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, and the in-storage per-unit-of-demand-and-time holding cost for each commodity at the different nodes is equal to 0.01.

Figure 1-(c) illustrates the 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 shown in Figure 1-(c) shows flow and fixed costs equal to 165 and 93, respectively. The three commodities are consolidated on arc $(2, 1)$, where commodities 1 and 3 wait 90 and 50 time

units before being consolidated with commodity 2, which arrives at node 2 at time 90. The holding cost is therefore equal to 42.5, thus resulting in a total solution cost equal to 300.5.

Figure 1-(d) illustrates the optimal solution for the SNDP with consideration of the holding costs. Solution 1-(d) shows flow and fixed costs equal to 165 and 96, respectively. The holding cost at node 2 is reduced from 42.5 to 22.5, since commodity 3 is routed on the alternative path $(3,1)$. The total cost of the solution is 283.5, being a cost saving equal to about 6% with respect to the total cost of solution 1-(c). Moreover, as shown in §EC.1 of the e-companion to this paper, such a cost saving percentage can be arbitrarily large in the best situation.

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.

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 $h_i^k \in \mathbb{R}_{>0}$. Hereafter, we also use the term *holding costs* to refer to both the *in-transit* and *in-storage* holding costs.

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 the current solution of the relaxation is no longer feasible for the corresponding 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 to solve the CTSNDP-HC.
- To ensure the convergence of the DDD algorithm, we propose and adopt a new refinement strategy.
- We validate the efficiency and effectiveness of the new algorithm through extensive computational experiments. The computational results also demonstrate the benefits that can be gained by taking into account 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, but also enhances the DDD algorithm and extends its practical 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. In §2, we review the relevant literature. We prove the existence of a complete TI model for the CTSNDP-HC in §3, followed by an overview of the DDD algorithm in §4. The new relaxation is described in §4.1, and the algorithm details, including a primal heuristic and a refinement strategy, are illustrated in §5. We provide the computational studies in §6. We conclude the paper and indicate future research directions in §7.

To help the reader throughout the rest of the paper, in §EC.2 of the e-companion to this paper we include a glossary of the symbols introduced.

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 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 also investigated by Dash et al. (2012) to solve the TSPTW by means of 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. 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-transit holding costs are equal to zero.

2.2. Handling consolidation costs

Consolidation is the process of grouping different items, originating at different locations and different times, into single vehicle loads at intermediate terminals or facilities, and transportation costs must be weighed against the penalties (handling and in-storage holding costs) that come with consolidation (Hall 1987).

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 nonnecessarily 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, $n = 1, \ldots, \eta^k$, satisfy the following system of inequalities:

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

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

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

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 $t^k_n + \tau_{a^k_n}$ 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 $a^k_n \in P^k$, $n = 1, 2, \ldots, \eta^k$, a departure time that corresponds to time t^k_n. 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.

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^k_n, $n = 1 \ldots, \eta^k + 1$, at the different nodes of the path that can be computed as follows:

$$\delta^k_n = \begin{cases} t^k_n - e^k, & n = 1, \\ t^k_n - (t^k_{n-1} + \tau_{a^k_{n-1}}), & n = 2, \ldots, \eta^k, \\ l^k - (t^k_{n-1} + \tau_{a^k_{n-1}}), & n = \eta^k + 1. \end{cases} \tag{2}$$

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 $a^k_n = \alpha_r$, 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 $\Theta(\alpha) = \{t^k_n : k \in \mathcal{K}, a^k_n \in P^k, a^k_n = \alpha\}$ 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 $\mathcal{W}$:

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

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:

$$z(\mathcal{W}) = \sum_{\mathcal{C}_r\in\mathcal{C}} f_{\alpha_r} \left\lceil \frac{\sum_{(k,n)\in J_r} q^k}{u_{\alpha_r}} \right\rceil + \sum_{k\in\mathcal{K}} \sum_{n=1}^{\eta^k} c^k_{a^k_n} q^k + \sum_{k\in\mathcal{K}} \sum_{n=1}^{\eta^k+1} (h^k_{v^k_n} q^k) \delta^k_n, \tag{3}$$

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

Table 1

k	P^k	$\mathcal{C}$	
		α	J
1	$(2,1)$	$(2,1)$	$\{(1,1),(2,2),(3,2)\}$
2	$(4,2,1)$	$(4,2)$	$\{(2,1)\}$
3	$(3,2,1)$	$(3,2)$	$\{(3,1)\}$

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 $\{t^k\}_{k\in\mathcal{K}}$, such that for each $k \in \mathcal{K}$, path P^k is k-feasible, and that from the set $\mathcal{C}$ of consolidation plans, the waiting times $\{\delta^k\}_{k\in\mathcal{K}}$ can be computed by expressions (2). 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 $(2,1)$, $(4,2)$, and $(3,2)$. In particular, all three commodities are consolidated on arc $\alpha = (2,1)$.

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_i^k be a nonnegative variable denoting the holding or waiting time of commodity k at node i. Formulation SND-HC($\mathcal{D}_\mathcal{T}^\Delta$) is as follows:

$$z(\mathcal{D}_\mathcal{T}^\Delta) = \min \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_\mathcal{T}^\Delta} f_{ij} y_{ij}^{t\bar{t}} + \sum_{k\in\mathcal{K}} \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_\mathcal{T}^\Delta} (c_{ij}^k q^k) x_{ij}^{kt\bar{t}} + \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}} (h_i^k q^k) w_i^k \tag{4}$$

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

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

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

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

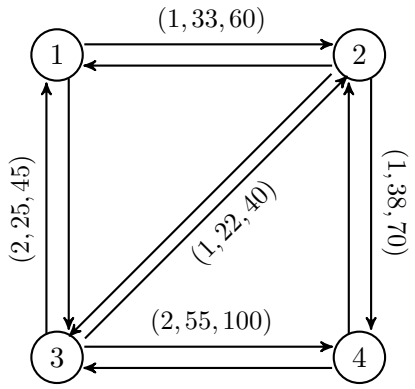
$$y_{ij}^{t\bar{t}} \in \mathbb{N}_{\ge 0}, \quad \forall ((i,t),(j,\bar{t})) \in \mathcal{A}_\mathcal{T}^\Delta, \tag{9}$$

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

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. Finally, constraints (8), (9) and (10) state the domains of the decision variables.

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 $\Delta\lceil \tau_{ij}/\Delta \rceil \ge \tau_{ij}$, 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.

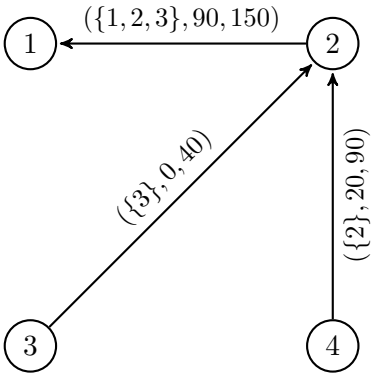
The existence of a value $\hat{\Delta}$ such that $z(\mathcal{D}_\mathcal{T}^{\hat{\Delta}})$ provides the optimal solution cost of the CTSNDP-HC is not straightforward. Indeed, as observed by Boland and Savelsbergh (2019), for some problems such as the TSPTW, a simple combinatorial argument suffices to show the existence of a complete TI model, whereas for other problems, such as the continuous-time inventory routing problem (Lagos et al. 2020), the discretization level may be smaller than one and difficult to identify.



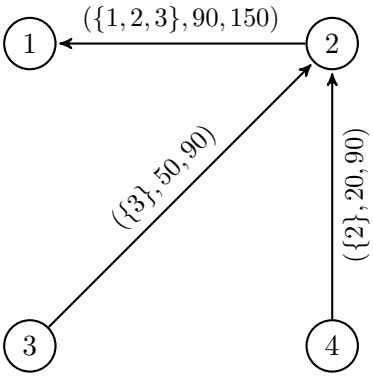
(a) Network $\mathcal{D}$: (c, f, τ) on arcs

k	o^k	d^k	q^k	e^k	l^k
1	2	1	25	0	160
2	4	1	30	20	180
3	3	1	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)

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 in G 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 the optimal solution of the CTSNDP. The alternative CTSNDP optimal solution represented by Figure 2-(d), where the departure time at the origin of commodities 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 $(1,2)$. When considering the CTSNDP-HC, if the unit holding costs for commodity 3 at terminals 2 and 3 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 $(3, 50)$, 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.

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.3.1 in the e-companion to this paper. □

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 $\Delta \in \mathbb{N}_{>0}$, $z(\mathcal{D}^{\Delta}_{\mathcal{T}})$ 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 $\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$ such that the optimal solution cost $z(\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}})$ of formulation SND-HC($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$) is the optimal solution cost of the CTSNDP-HC. However, the size of the network $\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$ can be prohibitively large, and the resulting TI model impractical to be solved by conventional techniques. For the CTSNDP, the DDD algorithm described by Boland et al. (2017) dynamically and iteratively determines the time points that are present in an optimal solution. The approach discovers which times are needed to obtain an optimal solution by solving a sequence of tractable integer programs. The exact algorithm described in this paper follows the DDD solution framework proposed by Boland et al. (2017), and adapts it in a novel way to solve the more general CTSNDP-HC.

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 $|\mathcal{N}_{\mathcal{T}}| \ll |\mathcal{N}^{\hat{\Delta}}_{\hat{\mathcal{T}}}|$. The algorithm starts by properly initializing the partial network $\mathcal{D}_{\mathcal{T}}$ (see §5.1), and at each iteration of the exact algorithm the following steps are executed:

(1) Solve a novel relaxation of formulation SND-HC($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$), thus computing a valid lower bound LB on the optimal solution cost $z(\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}})$. 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 (see §4.1).

(2) Compute a valid upper bound UB based on the flat solution $\mathcal{S}$ defined by the relaxation (see §5.2).

(3) Let $gap = (UB - LB)/UB$. If $gap \leq$ *optimality tolerance*, then terminate, where *optimality tolerance* is a user-defined parameter. If the *optimality tolerance* is set equal to zero, an optimal CTSNDP-HC solution has been identified. Otherwise, the solution corresponding to UB is within *optimality tolerance* of the optimal.

(4) Refine the network $\mathcal{D}_\mathcal{T}$ to include new time points and corresponding arcs (see §5.3).

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 of Step 4, which guarantee that the final relaxation model will eventually converge to the complete TI model (see §5.4).

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 $z(\mathcal{D}_{\hat{\mathcal{T}}}^{\hat{\Delta}})$.

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 $\bar{t} \leq t + \tau_{ij}$.*

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($\mathcal{D}_{\hat{\mathcal{T}}}^{\hat{\Delta}}$) with zero holding costs (i.e., an optimal CTSNDP solution) and let $\overline{\mathcal{A}} = \{((i,t),(j,t+\tau_{ij})) \in \mathcal{A}_{\hat{\mathcal{T}}}^{\hat{\Delta}} : \bar{y}_{ij}^{t,t+\tau_{ij}} > 0\}$ 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 $\rho_i(t) = \arg\max\{s \in \mathcal{T}_i : s \leq t\}$. 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 $t \leq e_k + \underline{\tau}_{o_k i}$. Further, by Property 3 a timed-copy arc $((i,\rho_i(t)),(j,t')) \in \mathcal{A}_\mathcal{T}$ of arc a exists in $\mathcal{D}_\mathcal{T}$ for some $(j,t') \in \mathcal{N}_\mathcal{T}$, and define $\sigma(a)$ to be any such t'.

The following proposition 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($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$).

Proposition 2 (Boland et al. (2017)) *If the in-storage holding costs h^k_i 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 $\tilde{a}=((i,\tilde{t}),(j,\tilde{t}'))\in\mathcal{A}_\mathcal{T}$:*

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

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 $\overline{P}^k=(a^k_1,\ldots,a^k_{\eta^k})$, $a^k_h\in\overline{\mathcal{A}}$, $h=1,\ldots,\eta^k$, induced by solution $(\bar{x},\bar{y})$ with $a^k_h=((i^k_h,t^k_h),(i^k_{h+1},t^k_h+\tau_{i^k_hi^k_{h+1}}))$ and $t^k_{h+1}\geq t^k_h+\tau_{i^k_hi^k_{h+1}}$ for $h=1,..,\eta^k-1$, there is a corresponding a feasible path $P^k=(\mu(a^k_1),\ldots,\mu(a^k_{\eta^k}))$ in $\mathcal{D}_\mathcal{T}$ with appropriate holding arcs.

The above proposition relies on the fact that adding additional holding arcs does not result in additional cost. In the presence of a nonzero holding cost, given an optimal $(\bar{x},\bar{y},\bar{w})$ solution of formulation SND-HC($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$) and corresponding paths $\{\bar{P}^k\}_{k\in\mathcal{K}}$, for each $k\in\mathcal{K}$ we have that:

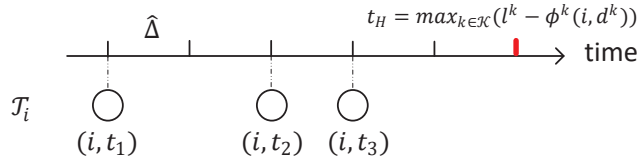
$$\bar{w}^k_{i^k_h}=\begin{cases}t^k_h-e^k, & h=1,\\ l^k-(t^k_{h-1}+\tau_{i^k_{h-1}i^k_h}), & h=\eta^k+1,\\ t^k_h-(t^k_{h-1}+\tau_{i^k_{h-1}i^k_h}), & \text{otherwise.}\end{cases}\quad\forall\ h=1,\ldots,\eta^k.$$

If we now consider the mapped solution (x,y,w) based on Proposition 2, since for each $k\in\mathcal{K}$ variables w depend only on variables x (see equations (7)) and $\rho_{i^k_h}(t^k_h)\leq t^k_h$ and $\sigma(a^k_h)\leq t^k_h+\tau_{i^k_hi^k_{h+1}}$, $\forall\ h=1,\ldots,\eta^k$, we have:

$$w^k_{i^k_h}=\begin{cases}\rho_{i^k_h}(t^k_h)-e^k\leq\bar{w}^k_{i^k_h}, & h=1,\\ l^k-\sigma(a^k_{h-1})\geq\bar{w}^k_{i^k_h}, & h=\eta^k+1,\\ \rho_{i^k_h}(t^k_h)-\sigma(a^k_{h-1})\geq\bar{w}^k_{i^k_h}\ \text{or}\ \rho_{i^k_h}(t^k_h)-\sigma(a^k_{h-1})\leq\bar{w}^k_{i^k_h}, & \text{otherwise,}\end{cases}\quad\forall\ h=1,\ldots,\eta^k.$$

Therefore, 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($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$).

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



(a)

τ_i	$\vec{t}_i(t)$	$\xi_i(t)$
t_1	t_2	t_2
t_2	t_3	t_2
t_3	t_H	t_H

(b)

Figure 3 Illustration of expression (15)

(i) On the partial network $\mathcal{D}^{\Delta}_{\mathcal{T}}$ satisfying Properties 1-3, with each arrival time $\bar{t}^k_h$, $h = 2, .., \eta^k + 1$, we can associate a lower bound $\check{t} \le \bar{t}^k_h$ computed as $\check{t} = \sigma(a^k_{h-1})$. In addition, with each departure time t^k_h, we can associate an upper bound $\hat{t} \ge t^k_h$ computed as $\hat{t} = \xi_{i^k_h}(t^k_h)$, so that we have $\rho_{i^k_h}(t^k_h) \le t^k_h \le \hat{t} = \xi_{i^k_h}(t^k_h)$. With these, we can obtain an upper bound on the holding time at node i_h, i.e., $t^k_h - \bar{t}^k_h \le \hat{t} - \check{t}$.

(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 $\mathcal{A} \setminus \{(i,o^k) : (i,o^k) \in \mathcal{A}\} \setminus \{(d^k,j) : (d^k,j) \in \mathcal{A}\}$ computed with respect to travel times τ_{ij}. Given any k-feasible path P^k and two nodes i and j visited by the path, where j follows i in the path, value $\phi^k(i,j)$ represents a valid lower bound on the difference between the arrival time at node j and the departure time at node i of path P^k.

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

$$\vec{t}_i(t) = \begin{cases} \arg\min\{t' \in \mathcal{T}_i : t' > t\}, & \text{if } t < \arg\max\{t' \in \mathcal{T}_i\}, \\ \max_{k\in\mathcal{K}}(l^k - \phi^k(i,d^k)), & \text{if } t = \arg\max\{t' \in \mathcal{T}_i\}, \end{cases} \tag{14}$$

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 o^k before l^k. Function $\xi_i(t)$ for node $i \in \mathcal{N}$ and time point $t \in \mathcal{T}_i$, can be computed as:

$$\xi_i(t) = \begin{cases} \vec{t}_i(t), & \text{if } \vec{t}_i(t) - t > \hat{\Delta}, \\ t, & \text{if } \vec{t}_i(t) - t \le \hat{\Delta}. \end{cases} \tag{15}$$

Figure 3 illustrates the computation of function $\xi_i(t)$.

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

$$z_R(\mathcal{D}_{\mathcal{T}}) = \min \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}} f_{ij} y^{t\bar{t}}_{ij} + \sum_{k\in\mathcal{K}} \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_{\mathcal{T}}} (c^k_{ij} q^k) x^{kt\bar{t}}_{ij} + \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}} (h^k_i q^k) w^k_i \tag{16}$$

$$s.t. \quad (5), (6), (8), (9), (10) \text{ and} \tag{17}$$

$$w_i^k \leq \begin{cases} \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_\mathcal{T}} \xi_i(t) x_{ij}^{k\bar{t}\bar{t}} - e^k, & i = o^k, \\ l^k - \sum_{((j,\bar{t}),(i,t))\in\mathcal{A}_\mathcal{T}} t\, x_{ji}^{k\bar{t}t}, & i = d^k, \\ \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_\mathcal{T}} \xi_i(t) x_{ij}^{k\bar{t}\bar{t}} - \sum_{((j,\bar{t}),(i,t))\in\mathcal{A}_\mathcal{T}} t\, x_{ji}^{k\bar{t}t}, & \text{otherwise}, \end{cases} \quad \forall\, i\in\mathcal{N}, \forall\, k\in\mathcal{K}, \tag{18}$$

$$\sum_{i\in\mathcal{N}} w_i^k = l^k - e^k - \sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_{\hat{\mathcal{T}}}} \tau_{ij} x_{ij}^{k\bar{t}\bar{t}}, \quad \forall\, k\in\mathcal{K}. \tag{19}$$

Inequalities (18) relax equations (7), whereas constraints (19) impose the total holding time of each commodity $k\in\mathcal{K}$. The following theorem holds.

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

Proof. The proof is provided in §EC.3.2 in the e-companion to this paper. □

The lower bound LB computed at Step 1 of the exact algorithm is thus computed as $LB = z_R(\mathcal{D}_\mathcal{T})$ where, in the computational results reported in §6, relaxation SND-HC-R($\mathcal{D}_\mathcal{T}$) is solved to optimist by means of a general MIP solver.

4.2. Strengthening relaxation SND-HC-R($\mathcal{D}_\mathcal{T}$)

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

We first observe that since variables w are nonnegative, equations (19) also imply that

$$\sum_{((i,t),(j,\bar{t}))\in\mathcal{A}_\mathcal{T}} \tau_{ij} x_{ij}^{k\bar{t}\bar{t}} \leq l^k - e^k, \quad \forall\, k\in\mathcal{K}, \tag{20}$$

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 $l^k - e^k$.

As discussed by Boland et al. (2017), the CTSNDP relaxation can be strengthened by means of the following additional property.

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 $t' < t'' \leq t + \tau_{ij}$, i.e., $t' = \arg max\{s : s \leq t+\tau_{ij}, (j,s)\in\mathcal{N}_\mathcal{T}\}$.*

The following theorem holds.

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 for all $D_\mathcal{T}$ satisfying Properties 1-3*

$$z_R(\overline{\mathcal{D}}_\mathcal{T}) \geq z_R(\mathcal{D}_\mathcal{T}). \tag{21}$$

Proof. The proof is provided in §EC.3.3 in the e-companion to this paper. □

5. Algorithm details

In this section, we describe in detail the different components of the DDD algorithm for the CTSNDP-HC. We start by describing how the partial time-expanded network is initialized (§5.1), followed by the details of the algorithm used to compute a valid upper bound (§5.2). Finally, we illustrate the refinement strategy (§5.3) and show that the exact algorithm described in §4 solves the CTSNDP-HC to optimality in a finite number of iterations (§5.4).

5.1. 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 $t' = \arg\max\{s \in \mathcal{T}_j : s \leq t + \tau_{ij}\}$ 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 $\mathcal{H}_\mathcal{T}$.

Function $\xi_i(t)$, $\forall i \in \mathcal{N}$, $t \in \mathcal{N}_\mathcal{T}$, is initialized based on the initial partially time-expanded network and expressions (15).

In order to keep the number of variables and constraints of formulation SND-HC-R($\mathcal{D}_\mathcal{T}$) as small as possible, we use the reduction rules described in Marshall et al. (2021). These rules are based on shortest path calculations and can identify the variables and constraints that cannot be part of any optimal SND-HC-R($\mathcal{D}_\mathcal{T}$) solution. The reader is referred to Marshall et al. (2021) for additional details.

5.2. Computing a feasible CTSNDP-HC solution

Step 2 of the exact algorithm (see §4) computes an upper bound UB on the optimal solution cost of CTSNDP-HC based on the implementable model $IM(\mathcal{S})$ described in §3.3. More precisely, the flat solution $\mathcal{S} = (\mathcal{P}, \mathcal{C})$ computed by solving relaxation SND-HC-R($\mathcal{D}_\mathcal{T}$) is used to derive a feasible CTSNDP-HC solution. Since SND-HC-R($\mathcal{D}_\mathcal{T}$) incorporates the holding cost, our upper bound heuristic method also incorporates the holding cost.

For the CTSNDP, where holding costs are all assumed to be zero, a solution $(\pi, t, \hat{t})$ satisfying constraints (12b)-(12i) together with the flat solution $\mathcal{S}$ corresponds to a feasible CTSNDP solution of cost $z_{fc}(\mathcal{S}) = LB$. Thus, as also observed by Boland et al. (2017), for the CTSNDP optimality is proved whenever the flat solution $\mathcal{S}$ associated with the relaxation is implementable. Hence, problem $IM(\mathcal{S})$ for the CTSNDP reduces to a feasibility problem. Marshall et al. (2021)

investigated the structure of the flat solution $\mathcal{S}$ for the CTSNDP as a graph theoretical problem, and identified two cases inducing non-implementable flat solutions (see Lemmas 1 and 2 of Marshall et al. (2021)).

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. Clearly, if $UB = LB$, then an optimal CTSNDP-HC solution has been identified.

For a given flat solution $\mathcal{S}$ associated with a feasible solution of relaxation SND-HC-R($\mathcal{D}_{\mathcal{T}}$), due to inequalities (20), 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). Below, we describe a procedure aimed at checking if the flat solution $\mathcal{S}$ is implementable. Moreover, if the solution is non-implementable, the procedure selectively identifies infeasible consolidation constraints with the aim of computing a valid upper bound UB of good quality.

The procedure performs the following two steps: (i) formulation $IM(\mathcal{S})$ is solved without constraints (12f) in order to identify infeasible consolidation constraints (12d) which are related to Lemma 1 of Marshall et al. (2021) and are selectively removed from the formulation; and (ii) the resulting updated $IM(\mathcal{S})$ model with constraints (12f) and a reduced set of constraints (12d) is solved in order to identify additional infeasible consolidation constraints which are related to Lemma 2 of Marshall et al. (2021).

The procedure 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). Algorithm 1 gives the steps of the procedure. In the algorithm, sets $\mathcal{J}$, $\mathcal{J}_C$ and $\mathcal{J}_P$ represent index sets associated with consolidation constraints (12d). Function `LP-Solve(.)` iteratively solves an LP model until a feasible solution is found. The function updates the set of consolidation constraints $\mathcal{J}$ and determines the set of infeasible consolidations $\overline{\mathcal{J}}$. Algorithm 1 first identifies the set $\mathcal{J}_C$ of infeasible consolidation constraints (line 13) and then the additional set of infeasible constraints $\mathcal{J}_P$ (line 16). The final UB value is computed based on the final subset of consolidation constraints identified.

Algorithm 1: Upper bound computation and identification of the sets of infeasible consolidation constraints

Input: Flat solution $\mathcal{S} = (\mathcal{P}, \mathcal{C})$
Output: Upper bound UB and consolidation constraint subsets $\mathcal{J}_C$ and $\mathcal{J}_P$

Function `Solve-LP(`*LP*, $\mathcal{J}$, $\overline{\mathcal{J}}$`)`**:**
  begin
    while *not solved* **do**
      Solve problem LP with consolidation constraints in $\mathcal{J}$;
      if *LP is infeasible* **then**
        Detect an IIS and select from the IIS $\mathcal{J}' \subseteq \mathcal{J}$ and compute $r^* = \arg\min\{f_{\alpha_r} : r \in \mathcal{J}'\}$;
        $\mathcal{J} \leftarrow \mathcal{J} \setminus \{r^*\}$ and $\overline{\mathcal{J}} \leftarrow \overline{\mathcal{J}} \cup \{r^*\}$;
      end
    end
    return $\mathcal{J}$ *and* $\overline{\mathcal{J}}$;
  end
begin
  `// Initialization`
  $\mathcal{J} \leftarrow \{1, \ldots, |\mathcal{C}|\}$, $\mathcal{J}_C \leftarrow \emptyset$ and $\mathcal{J}_P \leftarrow \emptyset$;
  `// Compute set` $\mathcal{J}_C$
  Let LP be model $IM(\mathcal{S})$ without constraints (12d) and (12f);
  `Solve-LP(`*LP*, $\mathcal{J}$, $\mathcal{J}_C$`)`;
  `// Update the set of consolidation constraints`
  $\mathcal{C} \leftarrow \{\mathcal{C}_r : r \in \mathcal{J}\}$;
  `// Compute set` $\mathcal{C}_P$
  Let LP be the model $IM(\mathcal{S})$ without constraints (12d);
  `Solve-LP(`*LP*, $\mathcal{J}$, $\mathcal{J}_P$`)`;
  `// Update the set of consolidation constraints`
  $\mathcal{C} \leftarrow \{\mathcal{C}_r : r \in \mathcal{J}\}$;
  `// A feasible solution has been identified`
  $UB \leftarrow z_{fc}(\mathcal{S}) + z_w(\mathcal{S})$;
  return UB, $\mathcal{J}_C$ *and* $\mathcal{J}_P$;
end

5.3. Refining a partially time-expanded network

If the exact algorithm does not terminate at Step 3 (see §4), then the upper bound UB computed by Algorithm 1 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 following two cases applies:

(i) The flat solution $\mathcal{S}$ is proved to be non-implementable, and at least one of the sets $\mathcal{J}_C$ and $\mathcal{J}_P$ is non-empty. This implies that there is at least one commodity $k \in \mathcal{K}$ routed on an arc $((i,t),(j,t'))$ that is too short, i.e., $t' < t + \tau_{ij}$. We call an arc $((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}}$ such that $t' < t + \tau_{ij}$ a *short-arc*. The reason for the non-implementability is due to the fact that short-arcs are evaluated by model $IM(\mathcal{S})$ with the actual or true travel times τ_{ij}. In this case, the short-arcs identified must be lengthened by adding new time points to network $\mathcal{D}_{\mathcal{T}}$. Network $\mathcal{D}_{\mathcal{T}}$ is also updated in such a way that the current solution of relaxation SND-HC-R($\mathcal{D}_{\mathcal{T}}$) is no longer feasible for the updated network $\mathcal{D}_{\mathcal{T}}$.

(ii) The relaxation of the holding variables w defined by inequalities (18) is too weak, that is, the upper bounds on the departure times computed by functions $\xi_i(.)$ and the lower bounds on the arrival times computed by functions $\sigma(.)$ must be strengthened.

We observe that if the flat solution $\mathcal{S}$ is implementable, a sufficient condition for the solution (x, y, w) of the relaxation SND-HC-R($\mathcal{D}_\mathcal{T}$) to correspond to an optimal CTSNDP-HC solution of cost LB is that $w_i^k = \theta_i^k$, $\forall k \in \mathcal{K}$, $i \in \mathcal{N}$, where:

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

If $w \neq \theta$, there exists at least one commodity k passing through a node i such that $w_i^k > \theta_i^k$, and at least one of the following two cases applies:

(a) The upper bounds on the departure times for node i computed by function $\xi_i(.)$ must be reduced. Let $((i,t),(j,t'))$ be an arc traversed by the path associated with commodity k, and let $b = \xi_i(t)$ be the time point computed with respect to the current $\mathcal{D}_\mathcal{T}$ network. To strengthen the relaxation, a new time $\bar{t}$ must be added to network $\mathcal{D}_\mathcal{T}$ such that the function $\xi_i(t)$ is updated in $\xi_i(t) = \bar{t} < b$. A time point $\bar{t} = (i, t + \max\{\lfloor (w_i^k - \theta_i^k)/2 \rfloor, 1\})$ suffices in this case.

(b) The lower bounds on the arrival times defined by the term $\sum_{((j',\bar{t}),(i,t))\in\mathcal{A}_\mathcal{T}} t\, x_{j',i}^{k\bar{t}t}$ of equations (22) for the cases $i \neq o^k$ must be increased. This can be accomplished by lengthening additional short-arcs (if any).

Clearly, for the CTSNDP-HC with zero in-storage holding costs, case (ii) does not apply, and in the worst case the set of nodes $\mathcal{N}_\mathcal{T}$ corresponds to set $\mathcal{N}_\mathcal{T}^\Delta$ with $\Delta = 1$.

In the next section, we describe how network $\mathcal{D}_\mathcal{T}$ can be updated with new time points based on the aforementioned cases (i) and (ii). §5.3.2 then discusses the strategy used to identify new time points.

5.3.1. Adding new time points to the partial time-expanded network $\mathcal{D}_\mathcal{T}$ The algorithm adopted to update the partial time-expanded network $\mathcal{D}_\mathcal{T}$ with a new time point (i,t) follows the steps of a similar algorithm used by the DDD algorithm for the CTSNDP by Boland et al. (2017). Algorithm 2 gives the steps performed to update a network $\mathcal{D}_\mathcal{T}$ with a new time point (i,t). In the algorithm, point (i,t) is first added to the set of nodes $\mathcal{N}_\mathcal{T}$ (line 2). The longest timed-copy arcs to node (i,t) are then added to the network in order to satisfy Properties 3 and 4 (line 5). Whenever an arc with which Property 4 is not satisfied because of the addition of node (i,t), the network is updated by removing the arcs not satisfying Property 4 and by adding new arcs that do satisfy Property 4 (line 10). Finally, the set of holding arcs is updated (lines 13-15).

Notice that the values of functions $\xi_i(.)$ change according to the updated network $\mathcal{D}_\mathcal{T}$ and expressions (15).

Algorithm 2: Refinement of network $\mathcal{D}_\mathcal{T}$ with a new time point

Input: Network $\mathcal{D}_\mathcal{T}$ and time point t for node i
Output: Updated network $\mathcal{D}_\mathcal{T}$

1 **Function** `Refine(`$\mathcal{D}_\mathcal{T}$, i, t`):`
  begin
2 $\mathcal{N}_\mathcal{T} \leftarrow \mathcal{N}_\mathcal{T} \cup \{(i,t)\}$;
  `// Add timed-copy arcs to node` (i,t) `satisfying Property 4`
3 **forall** $(i,j) \in \mathcal{A}$ **do**
4 $t' \leftarrow \arg\max\{s \in \mathcal{T}_j : s \leq t + \tau_{ij}\}$;
5 $\mathcal{A}_\mathcal{T} \leftarrow \mathcal{A}_\mathcal{T} \cup ((i,t),(j,t'))$;
6 **end**
  `// Change the arcs to impose Property 4`
7 **forall** $((j,\bar{t}),(i,t')) \in \mathcal{A}$ *where* $t' = \arg\max\{s \in \mathcal{T}_i : s < t\}$ **do**
8 **if** $\bar{t} + \tau_{ij} \geq t$ **then**
9 $\mathcal{A}_\mathcal{T} \leftarrow \mathcal{A}_\mathcal{T} \setminus \{((j,\bar{t}),(i,t'))\}$;
10 $\mathcal{A}_\mathcal{T} \leftarrow \mathcal{A}_\mathcal{T} \cup \{((j,\bar{t}),(i,t))\}$;
11 **end**
12 **end**
  `// Update holding arc set`
13 $t' \leftarrow \arg\max\{s \in \mathcal{T}_i : s < t\}$ and $t'' \leftarrow \arg\min\{s \in \mathcal{T}_i : s > t\}$;
14 $\mathcal{H}_\mathcal{T} \leftarrow \mathcal{H}_\mathcal{T} \setminus \{((i,t'),(i,t''))\}$;
15 $\mathcal{H}_\mathcal{T} \leftarrow \mathcal{H}_\mathcal{T} \cup \{((i,t'),(i,t)),((i,t),(i,t''))\}$;
16 **return** $\mathcal{D}_\mathcal{T}$;
  end
17

5.3.2. Refinement strategy The refinement strategy applied at Step 4 of the DDD Algorithm identifies new time points to update the network $\mathcal{D}_\mathcal{T}$ in order to strengthen the relaxation. It is based on sets $\mathcal{J}_C$ and $\mathcal{J}_P$ of infeasible consolidation constraints obtained from the computation of a valid upper bound by Algorithm 1.

An overview of the steps executed by the refinement strategy is given by Algorithm 3. In the algorithm, short-arcs associated with the set of infeasible consolidations $\mathcal{J}_C$ are first identified (*Strategy 1*, lines 1-8), followed by lengthening of the short-arcs associated with infeasible consolidations from set $\mathcal{J}_P$ (*Strategy 2*, lines 9-17). To strengthen the relaxation of the holding variables w, the algorithm lengthens the short-arcs associated with paths $\{P^k\}_{k\in\mathcal{K}}$ selected in the flat solution (*Strategy 3*, lines 18-23). It is worth noting that the paths composing an implementable flat solution can contain short-arcs, hence in this case Strategies 1 and 2 are not used since $\mathcal{J}_C = \mathcal{J}_P = \emptyset$, and the short-arcs are lengthened by Strategy 3. Finally, new time points are added by *Strategy 4* (lines 24-31).

5.4. Convergence and optimality

Given the domains of the x and y variables of formulation SND-HC($\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$) defined on the fully time-expanded network $\mathcal{D}^{\hat{\Delta}}_{\hat{\mathcal{T}}}$, there are only a finite number of solutions corresponding to all feasible solution vectors (x,y) of the formulation.

Algorithm 3: Refinement strategy

Input: Network $\mathcal{D}_\mathcal{T}$ and consolidation constraint sets $\mathcal{J}_C$ and $\mathcal{J}_P$
Output: Updated network $\mathcal{D}_\mathcal{T}$
begin
  `// Strategy 1: Lengthen short-arcs based on set` $\mathcal{J}_C$
  forall $r \in \mathcal{J}_C$ **do**
    forall $(k,n) \in J_r$ **do**
      $P^k \leftarrow$ path associated with commodity k;
      forall $((i,t),(j,t')) \in P^k$ **do**
        if $t' < t + \tau_{ij}$ **then** `Refine(`$\mathcal{D}_\mathcal{T}$`,`j`,`$t+\tau_{ij}$`)`;
      end
    end
  end
  `// Strategy 2: Lengthen short-arcs based on set` $\mathcal{J}_P$
  forall $r \in \mathcal{J}_P$ **do**
    forall $(k,n) \in J_r$ **do**
      $P^k \leftarrow$ path associated with commodity k;
      $((i,t),(j,t')) \leftarrow a^k_n$;
      if $t' < t + \tau_{ij}$ **then** `Refine(`$\mathcal{D}_\mathcal{T}$`,`j`,`$t+\tau_{ij}$`)` ;
      $((i,t),(j,t')) \leftarrow \arg\max_{((i_1,t_1),(i_2,t_2)) \in P^k} \{t_1 + \tau_{i_1,i_2} - t_2\}$;
      if $t' < t + \tau_{ij}$ **then** `Refine(`$\mathcal{D}_\mathcal{T}$`,`j`,`$t+\tau_{ij}$`)`;
    end
  end
  `// Strategy 3: Lengthen short-arcs`
  Sort the arcs $a = ((i,t),(j,t'))$ associated with $\{P^k\}_{k\in\mathcal{K}}$ having $t' < t + \tau_{ij}$ for increasing values of $\gamma_a = t$;
  $\mathcal{L} \leftarrow (a_1, a_2, \ldots, a_{|\mathcal{A}_\mathcal{T}|})$ such that $\gamma_{a_1} \leq \gamma_{a_2} \leq \cdots \leq \gamma_{a_{|\mathcal{A}_\mathcal{T}|}}$;
  forall $h \leftarrow 1$ *to* $\min\{|\mathcal{L}|, 50\}$ **do**
    $((i,t),(j,t')) \leftarrow a_h$;
    `Refine(`$\mathcal{D}_\mathcal{T}$`,`j`,`$t+\tau_{ij}$`)`
  end
  `// Strategy 4: Add new time points`
  forall $k \in \mathcal{K}$ **do**
    $P^k \leftarrow$ path associated with commodity k;
    forall $((i,t),(j,t')) \in P^k$ **do**
      if $w^k_i > \theta^k_i$ **then**
        `Refine(`$\mathcal{D}_\mathcal{T}$`,`j`,`$t + \max\{\lfloor (w^k_i - \theta^k_i)/2 \rfloor, 1\}$`)`;
      end
    end
  end
end

At the different iterations of the algorithm, if a flat-solution is non-implementable, Algorithm 1 identifies at least one short-arc which is lengthened by refinement Strategies 1 and 2. Moreover, if the current relaxation of the holding variables w is too weak, new additional short-arcs are lengthened and new time points are added to strengthen the relaxation by refinement Strategies 3 and 4, respectively. As a result of the refinement strategies, the current solution of the relaxation is no longer feasible for the updated relaxation SND-HC-R($\mathcal{D}_\mathcal{T}$).

Under the assumption that a feasible solution exists, since there are only finitely many solutions, arcs to be lengthened and time points to be added, repeating this process finitely many times will guarantee the convergence of the solution associated with lower bound LB to an optimal solution of 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 executions of Steps 1-4.*

It is worth noting that with zero in-storage holding costs, refinement Strategies 1 and 2 suffice for the convergence of the algorithm, since case (ii) of Section 5.3 does not apply.

6. Computational experiments

In this section, we present extensive experimental analysis, with two main aims. Firstly, we evaluate the performance of the DDD algorithm for the CTSNDP-HC on instances derived from the literature (see §6.1). Secondly, we analyze the effectiveness of the algorithm by the quality of the solutions produced on an additional set of instances, and we identify the factors that affect the complexity of the CTSNDP-HC (see §6.2). Moreover, we evaluate the benefits that can be gained by taking into account holding costs when solving the CTSNDP.

The algorithm was 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($\mathcal{D}_{\mathcal{T}}$) at Step 1 of the exact algorithm described in §4. The Gurobi function `Model.computeIIS()` was used to compute IISs in Algorithm 1. 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.

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. Algorithm EXM-0 will be used as a baseline algorithm to compute heuristic solutions for the CTSNDP-HC and to analyze the performance of algorithm EXM. 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* used at Step 3 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. Given tolerance tol_{MIP} applied to the MIP solver and in order to compute safe lower bounds, at Step 1 of EXM 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 `ObjBoundC`.

Table 2 Instances from Crainic et al. (2001)

Class	$\|\mathcal{N}\|$	$\|\mathcal{A}\|$	$\|\mathcal{K}\|$	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

6.1. Experiments based on CTSNDP benchmark instances

For our first set of experiments, we tested the performance of EXM in solving CTSNDP and CTSNDP-HC instances generated from CTSNDP benchmark instances used in the literature.

6.1.1. Instance Generation 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.

The instances were derived from the "C" instances presented in Crainic et al. (2001) which have been used as benchmark instances for the capacitated fixed charge network design problem. Table 2 gives the details of the 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 class 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 first computing for each node $i \in \mathcal{N}$ parameter $\alpha_i = 1/|\mathcal{A}_i| \sum_{a \in A_i} (c_a + f_a/u_a)/\tau_a$, where $\mathcal{A}_i = \{(j_1, i), (i, j_2) : (j_1, i), (i, j_2) \in \mathcal{A}\}$ represents the average fixed and variable costs of all the ingoing and outgoing arcs of node i. Then, the per-unit-of-demand-and-time cost h_i^k was set equal to $0.3\,\alpha_i$, $\forall k \in \mathcal{K}$. Value 0.3 was chosen to simulate the fact that holding costs typically range between 20% and 30% of the inventory value (Needham and Evers 1998, Rudi et al. 2016), and the transportation cost is generally less than the commodity value. Since in practice a commodity does not incur any holding costs at the destination, for each commodity $k \in \mathcal{K}$ we set $h_{d^k}^k = 0$. We therefore generated 558 CTSNDP-HC instances.

6.1.2. Results In this section we analyze the performances of EXM in solving both CTSNDP and CTSNDP-HC instances, and we also report on a comparison of the results obtained by algorithm EXM-0 with the results of the DDD algorithms proposed by Boland et al. (2017), denoted as BHMS17, and Marshall et al. (2021), denoted as MBSH21.

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.

We first summarize the results obtained by the baseline algorithm EXM-0 in solving the 558 CTSNDP instances, and we then compare its performances 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. 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.

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. Regarding EXM-0, tol_{DDD} and tol_{MIP} were set equal to 0.01 and 0.01, respectively.

Based on Boland and Savelsbergh (2019) and as also reported by Marshall et al. (2021), the results are 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".

Table 3 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. For algorithms BHMS17 and MBSH21, value LB corresponds to the lower bound computed at the last iteration of the DDD algorithm.

For method EXM-0, due to the use of the tol_{MIP} tolerance and the computation of the corresponding safe lower bound, 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 performances on the different 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 3, we adopted algorithm EXM-0 as a baseline algorithm for comparison purposes with EXM.

For the set of 558 CTSNDP-HC instances, we executed EXM with a time limit of two hours, and we also used EXM-0 to solve the corresponding CTSNDP instances and to compute heuristic solutions for the CTSNDP-HC, again with a time limit of two hours. In order 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. For these experiments, we used $tol_{DDD}=0.01$ for both EXMand EXM-0, $tol_{MIP}=0.01$ for EXM-0. For algorithm EXM tol_{MIP} is also computed dynamically as $\max\{\min\{0.04, gap\times 0.25\}, 0.01\}$.

Table 3 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	0.78	318.5	12.1	95.6
HC/HF	BHMS17	0.56	1966.7	6.0	53.7
177	MBSH21	0.84	1693.8	17.5	56.5
	EXM-0	3.31	1613.2	11.6	60.5
LC/LF	BHMS17	0.00	28.6	3.7	100.0
94	MBSH21	0.00	0.6	6.5	100.0
	EXM-0	0.62	0.8	3.7	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	0.50	0.1	1.5	100.0

Table 4 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	2.4	5.2	3.7	450.6	88.6	12.2	93.4	1.0	2.8	1.7	1170.9	93.3	7.5	78.1	1.0	4.5	1.7	2544.0	94.1	15.9	1.1	0.7	2.5
HC/HF	70.6	1.4	23.6	7.8	2878.0	95.1	12.7	68.9	1.2	8.8	2.9	3125.5	97.7	7.6	55.9	1.2	10.3	3.6	3786.3	97.7	10.0	1.7	1.5	14.5
LC/LF	100.0	-	-	-	0.8	60.0	3.7	100.0	-	-	-	1.3	68.0	3.1	100.0	-	-	-	1.9	70.7	4.7	0.5	1.3	2.8
LC/HF	100.0	-	-	-	0.1	40.9	1.5	100.0	-	-	-	0.1	51.7	1.6	100.0	-	-	-	0.2	55.2	2.4	0.2	1.5	3.1

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

Table 4 reports the corresponding results. 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, obtained by adding the holding costs associated with the holding times of the solution corresponding to $UB0$.
- 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 5 gives a different view of the results of Table 4 by grouping the instances based on the grouping of Table 2. In the table, column "*ni*" gives the number of instances in the corresponding group and "*opt*" is the number of instances solved to optimality.

Table 4 indicates that EXM achieves a similar performance to that of EXM-0 when used to solve the CTSNDP. The detailed Table 5 shows that EXM solved 491 instances to optimality, whereas 500 instances were solved by EXM-0 but that there are instances solved to optimality by EXM that cannot be solved by EXM-0 within the imposed time limit, and vice versa. Interestingly, on the one hand EXM requires (on average) a smaller number of iterations to converge to optimality than EXM-0, thus showing the importance of the refinement strategy in DDD algorithms. On the other hand, EXM requires a higher computing time, as shown by column $\%tLB$. For both EXM and EXM-0, most of the computing time is taken up by the MIP solver used to compute the lower bounds at the different iterations. These results show that tuning a DDD algorithm requires finding the right trade-off between the time spent by the MIP solver and the rate of convergence of the DDD method, which strictly depends on the refinement strategy.

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 must be achieved.

Table 5 Summary results on the CTSNDP-HC instances by instances name

		Zero holding costs								Nonzero holding costs						
		EXM-0				EXM				EXM						
			%$UB0$				%UB				%UB			%$LB0$	%$UB1$	
Name	*ni*	*opt*	*min*	*max*	*avg*	*opt*	*min*	*max*	*avg*	*opt*	*min*	*max*	*avg*	*avg*	*avg*	*max*
c33	18	18	-	-	-	18	-	-	-	18	-	-	-	0.6	1.8	3.1
c35	18	18	-	-	-	18	-	-	-	18	-	-	-	0.1	1.6	2.8
c36	18	18	-	-	-	18	-	-	-	18	-	-	-	0.1	1.2	2.3
c37	18	18	-	-	-	16	1.8	2.4	2.1	11	1.5	3.9	2.6	1.3	0.0	0.7
c38	18	10	1.4	22.3	8.9	11	1.3	8.6	3.3	9	2.0	9.4	4.5	1.3	2.3	13.5
c39	18	15	2.9	8.8	5.6	16	1.6	4.0	2.8	9	1.2	3.7	2.5	1.6	0.6	5.5
c40	18	9	11.8	23.6	17.9	9	3.0	8.8	4.8	9	4.2	10.3	6.6	2.4	5.7	14.5
c41	18	18	-	-	-	18	-	-	-	18	-	-	-	0.2	1.5	2.9
c42	18	18	-	-	-	18	-	-	-	18	-	-	-	0.6	1.3	1.9
c43	18	18	-	-	-	18	-	-	-	18	-	-	-	0.2	1.8	2.7
c44	18	18	-	-	-	18	-	-	-	18	-	-	-	-0.1	0.9	1.2
c45	18	18	-	-	-	18	-	-	-	18	-	-	-	1.6	0.5	0.8
c46	18	14	1.8	6.8	4.1	14	1.2	2.6	1.7	11	1.6	3.2	2.0	1.4	0.8	3.9
c47	18	18	-	-	-	18	-	-	-	18	-	-	-	1.6	0.6	0.8
c48	18	13	4.6	16.5	9.6	12	1.9	3.5	2.9	9	1.2	7.7	3.0	1.7	1.9	10.1
c49	18	18	-	-	-	18	-	-	-	18	-	-	-	0.6	1.5	2.8
c50	18	18	-	-	-	18	-	-	-	18	-	-	-	1.2	1.1	2.1
c51	18	18	-	-	-	18	-	-	-	18	-	-	-	0.6	1.2	1.8
c52	18	18	-	-	-	18	-	-	-	18	-	-	-	1.0	1.2	2.0
c53	18	18	-	-	-	16	1.0	1.1	1.1	9	1.3	1.6	1.4	1.6	0.6	0.8
c54	18	10	1.7	5.5	3.4	10	1.8	3.1	2.4	9	2.7	4.1	3.5	1.3	0.6	2.6
c55	18	18	-	-	-	17	1.0	1.0	1.0	9	1.4	1.7	1.5	1.7	0.8	1.1
c56	18	13	2.3	4.3	3.4	10	1.2	3.5	1.9	9	2.3	4.5	3.2	1.5	0.5	2.3
c57	18	18	-	-	-	18	-	-	-	18	-	-	-	0.6	1.1	1.8
c58	18	18	-	-	-	18	-	-	-	18	-	-	-	0.7	0.9	1.6
c59	18	18	-	-	-	18	-	-	-	18	-	-	-	0.5	1.3	1.9
c60	18	18	-	-	-	18	-	-	-	18	-	-	-	0.7	1.3	2.3
c61	18	18	-	-	-	17	1.0	1.0	1.0	10	1.0	1.3	1.2	1.3	0.6	0.8
c62	18	9	1.9	8.0	5.4	10	1.7	3.4	2.8	9	3.6	5.2	4.3	1.5	1.2	3.4
c63	18	18	-	-	-	18	-	-	-	12	1.1	1.4	1.2	1.5	0.8	1.0
c64	18	11	1.8	5.2	3.4	9	1.1	2.8	2.0	9	2.2	4.5	3.1	1.4	0.7	2.5

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

The summary results of Table 4 and the detailed results of Table 5 show that the lower bound $LB0$ is quite tight, and that the upper bound $UB1$ (derived from the CTSNDP) is on average also quite tight, but that on the most difficult instances of group **HC/HF** a significant deviation from the optimal solution exists, as shown by the maximum percentage deviation being equal to 14.5%.

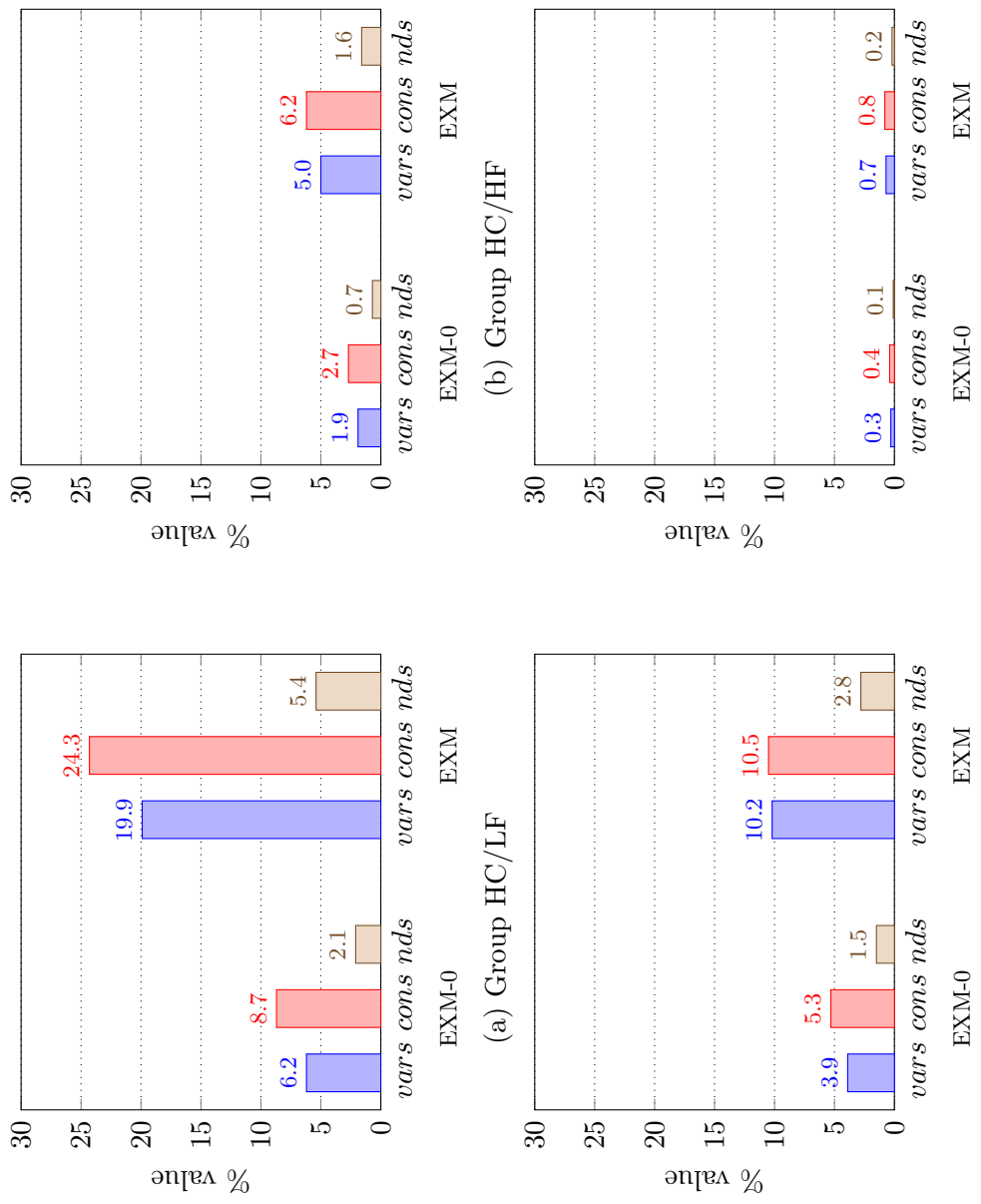


Figure 4 Comparison of partially and fully time-expanded networks

Table 6 CTSNDP-HC instances: holding costs, holding times and consolidations

		$\%hc$	$\%ht$	$\%cs$
HC/LF	$UB1$	1.9	10.3	47.7
	UB	1.2	7.0	47.5
HC/HF	$UB1$	2.3	13.7	59.9
	UB	1.7	10.4	61.3
LC/LF	$UB1$	1.8	7.7	14.5
	UB	0.5	2.3	13.0
LC/HF	$UB1$	1.8	7.3	9.3
	UB	0.3	1.5	8.0

In particular, Table 5 reports a negative $\%LB0$ value, showing that in some cases, due to the computation of a safe lower bound, a slightly better lower bound is achieved in solving the CTSNDP.

To analyze the effectiveness of the DDD approach, Figure 4 shows the relative sizes (average values), in terms of the number of variables ("*vars*") and constraints ("*cons*") of the relaxation models associated with the final partially time-expanded networks $\mathcal{D}_{\mathcal{T}}$ of 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 (or nodes) ("$\%nds$") of the partial network over the full network.

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, constraints and time points of EXM-0.

Table 6 summarizes relevant details of the solutions corresponding to the upper bounds $UB1$ and UB. More specifically, the table reports 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) $\%cs$, the number of consolidation arcs over the total number of arcs used by the solution.

The table 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 the EXM-0. Interestingly, this reduction is achieved by increasing the percentages of consolidations (see column $\%cs$) for group **HC/HF**, and by decreasing of the percentages for the remaining groups.

6.2. Experiments on newly generated CTSNDP-HC benchmark instances

In this section, we describe and evaluate a new set of CTSNDP-HC instances with two aims. First, we investigate attributes of the instances that affect the complexity of the problem by

exploiting two main components characterizing service network design problems: the connectivity of the underlying physical network (spatial component) and the flexibility of the shipments' time requirements (temporal component). Second, we analyze the benefits gained by incorporating holding costs into the CTSNDP.

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

(1) **Varying the connectivity level**. For each instance of Table 2 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 also compute time Γ 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 $\mathcal{P} = \{(i, j) : \underline{\tau}_{ij} \leq \Gamma\}$.

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 $\mathcal{P}$, 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 $\frac{1}{6} l_{avg}$.

Table 7 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

(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 $e^k + \underline{\tau}_{o^k d^k} + \alpha$ where α is the value drawn from a distribution.

Based on the above two steps, for each of the instances in Table 2, we generate four different networks $\mathcal{D}_1$, $x = 1, 2, 3, 4$, 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.

Table 7 summarizes the connectivity properties of the instances generated. Given an instance and the associated network $\mathcal{D}_x$, $x \in \{1, 2, 3, 4\}$, we first compute 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 then we compute $\min_{k \in \mathcal{K}}\{|(S, \mathcal{N} \setminus S)_k|\}$, i.e., the cardinality of the cut having the minimum cardinality among the different $o^k - d^k$ pair. For each type of network, the table reports 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.

6.2.2. Results For these experiments, a time limit of two hours was imposed on EXM-0 and EXM. Also, for these tests we used $tol_{DDD} = 0.01$ for both EXM and EXM-0, and $tol_{MIP} = 0.01$ for EXM-0, whereas for EXM, the tol_{MIP} was computed dynamically as $\max\{\min\{0.04, gap \times 0.25\}, 0.01\}$.

Table 8 summarizes the results obtained using the same notation introduced in Table 4. 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 of $\%LB0$ and $\%UB1$ are computed over all instances.

We recall that an increasing 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 flexibility and connectivity levels are particularly difficult. One reason for this complexity is the difficulty of the MIP problems

Table 8 Summary results on the new CTSNDP-HC instances

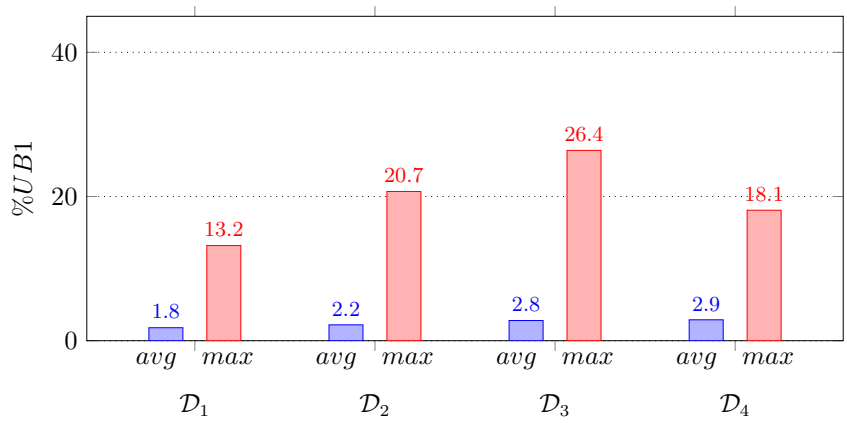
		Zero holding costs							Nonzero holding costs							
		EXM-0							EXM							
			%$UB0$							%UB						%$LB0$
Dist.	Network	%*opt*	*min*	*max*	*avg*	*time*	%tLB	*iter*	%*opt*	*min*	*max*	*avg*	*time*	%tLB	*iter*	*avg*
A	$\mathcal{D}_1$	86.0	1.1	18.1	7.0	1372.2	81.1	6.9	84.9	1.0	7.0	2.7	1577.8	88.2	7.5	2.0
	$\mathcal{D}_2$	86.0	1.6	25.1	9.7	1448.9	79.5	5.9	76.3	1.0	6.0	2.6	1919.7	86.2	6.8	2.2
	$\mathcal{D}_3$	82.8	1.1	30.2	9.1	1541.0	77.7	5.9	79.6	1.0	3.5	2.0	1853.5	83.6	7.3	2.5
	$\mathcal{D}_4$	86.0	1.4	24.2	10.4	1176.2	76.5	6.0	84.9	1.1	6.3	2.5	1456.4	82.7	7.9	2.7
B	$\mathcal{D}_1$	55.9	1.4	46.7	19.1	4389.6	89.1	3.6	55.9	1.0	14.2	4.7	3384.9	93.9	6.3	3.7
	$\mathcal{D}_2$	53.8	1.1	47.5	18.1	4535.6	86.2	3.5	57.0	1.2	13.1	4.5	3416.6	92.2	6.2	3.9
	$\mathcal{D}_3$	55.9	2.6	37.0	14.4	4418.3	80.4	3.0	53.8	1.0	17.9	4.2	3596.4	88.1	6.6	3.5
	$\mathcal{D}_4$	74.2	2.6	41.8	13.5	2650.3	78.1	3.9	67.7	1.0	13.7	3.7	2719.4	85.0	8.0	3.2
C	$\mathcal{D}_1$	47.3	1.1	54.7	21.6	3892.0	93.0	3.0	45.2	1.4	40.1	8.1	3986.3	97.2	5.5	4.0
	$\mathcal{D}_2$	48.4	1.3	57.1	19.9	3769.7	90.6	2.8	46.2	1.2	45.3	7.9	3976.2	95.9	5.7	3.9
	$\mathcal{D}_3$	51.6	2.0	39.8	15.2	3651.2	86.3	3.0	47.3	1.0	19.8	5.7	3859.0	93.5	5.9	3.7
	$\mathcal{D}_4$	66.7	1.3	42.8	12.6	2592.8	80.8	3.7	62.4	1.1	15.8	4.5	3118.5	88.4	8.0	3.3

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 by the MIP solver and by the number of iterations, the entire solution process is slowed down.

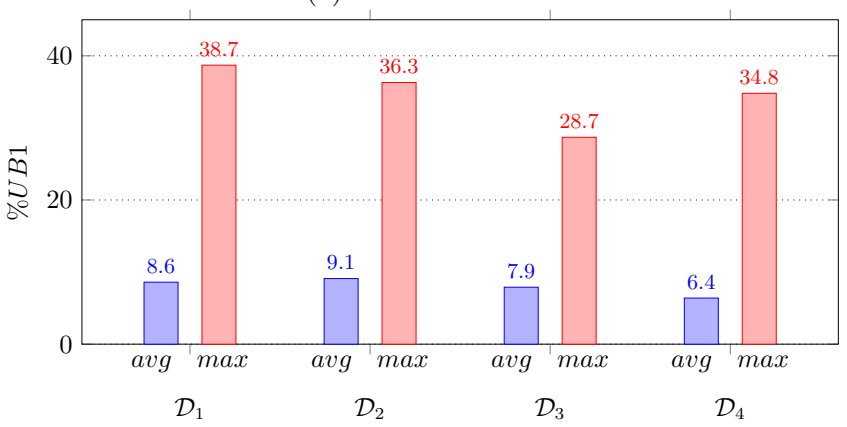
The percentage deviations of the lower bound $LB0$ clearly show that the higher the flexibility, the higher is the deviation, that is, many more consolidation opportunities are allowed but the corresponding holding costs cannot be ignored. In this case, solving the CTSNDP and, based on the corresponding solutions, deriving corresponding CTSNDP-HC solutions, is clearly not a valid option. Indeed, as shown by Figure 5, which shows the average and maximum percentage deviations of upper bound $UB1$, the cost saving percentage of the minimal total cost can be up to 40.8%.

Table 9 gives the relative sizes of the partial networks over the fully time-expanded networks of EXM-0 and EXM. In particular, the table reports results about the instances solved to optimality by both algorithms. Also in the table, the instances are grouped by distribution and network types, and the notation adopted is that of Figure 4. In addition, the table also shows percentage values concerning the number of consolidations ("%cs") associated with upper bounds $UB1$ and UB. Finally, Figure 6 depicts the ratios of the holding costs and holding times, again of the instances solved to optimality.

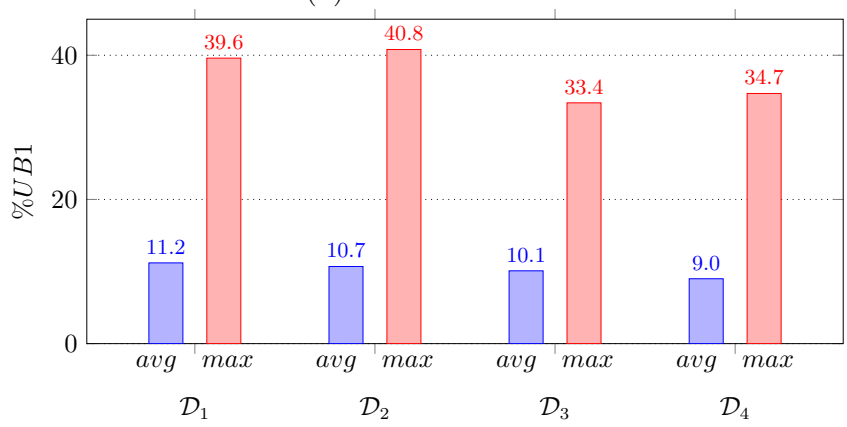
The results concerning the percentages of the number of consolidations (column "%cs") given by the table and the percentages of the holding times and costs depicted by the figure, show that significant reductions in the two measures are achieved by upper bound UB with respect to upper bound $UB1$, again depending on the connectivity of the underlying physical network and the flexibility of the shipments' time requirements. In addition to cost reduction, reducing the number of consolidations produces more reliable solutions. Indeed, in practice, routing issues that



(a) Distribution A



(b) Distribution B

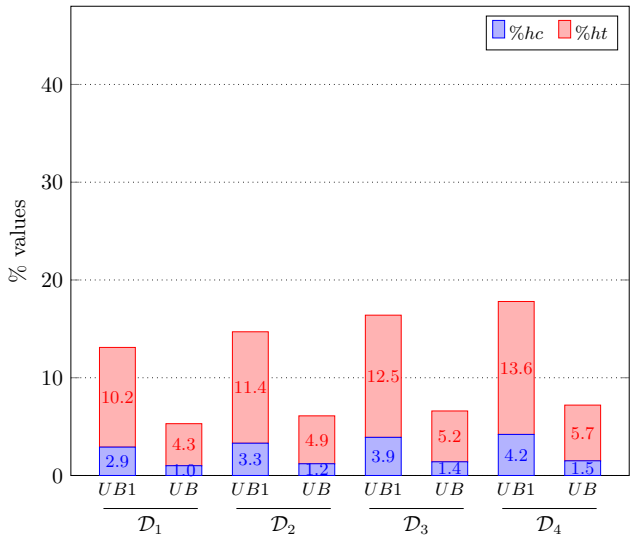


(c) Distribution C

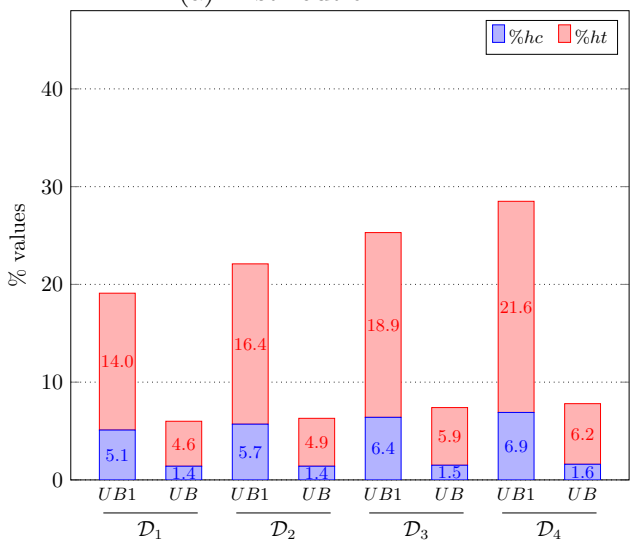
Figure 5 Results about upper bound $UB1$ showing the average and maximum cost saving percentages gained by the CTSNDP-HC solutions over CTSNDP-based solutions

may occur along a trip often cause delays to subsequent consolidations, which result in late services and deliveries.

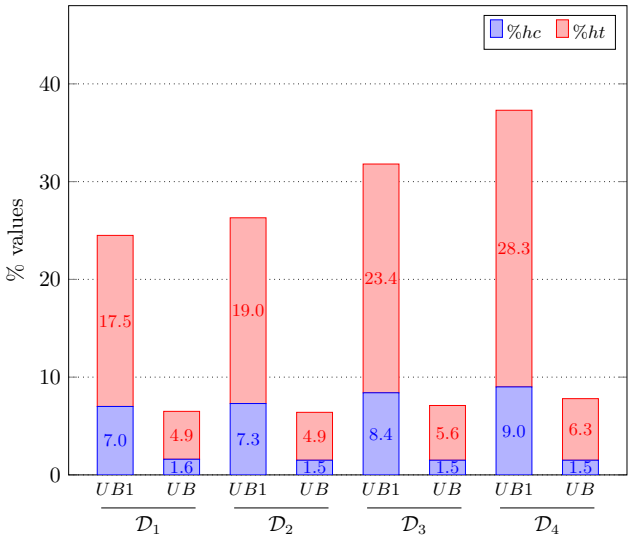
As observed for the first set of instances, Table 9 confirms the effectiveness of the DDD approach in confining the sizes of the different MIPs to small portions of the fully time-expanded model.



(a) Distribution A



(b) Distribution B



(c) Distribution C

Figure 6 Ratios of the holding costs and holding times

Table 9 Sizes of the partially and fully time-expanded networks and consolidations

Dist.	Network		$\%vars$	$\%cons$	$\%nds$	Upp. bounds	$\%cs$
A	$\mathcal{D}_1$	EXM-0	1.7	2.3	0.8	$UB1$	40.7
		EXM	3.8	4.3	1.4	UB	38.0
	$\mathcal{D}_2$	EXM-0	1.6	2.1	0.7	$UB1$	42.9
		EXM	3.4	3.9	1.3	UB	41.1
	$\mathcal{D}_3$	EXM-0	1.4	1.9	0.7	$UB1$	49.6
		EXM	2.9	3.3	1.1	UB	43.8
	$\mathcal{D}_4$	EXM-0	1.3	1.8	0.7	$UB1$	51.0
		EXM	2.5	3.2	1.2	UB	47.3
B	$\mathcal{D}_1$	EXM-0	1.0	1.4	0.6	$UB1$	49.4
		EXM	2.4	3.1	1.1	UB	41.3
	$\mathcal{D}_2$	EXM-0	0.9	1.3	0.6	$UB1$	53.3
		EXM	2.2	2.7	1.0	UB	42.2
	$\mathcal{D}_3$	EXM-0	0.8	1.1	0.5	$UB1$	55.4
		EXM	2.0	2.4	0.9	UB	46.5
	$\mathcal{D}_4$	EXM-0	0.7	1.0	0.6	$UB1$	57.8
		EXM	1.7	2.1	1.0	UB	47.8
C	$\mathcal{D}_1$	EXM-0	0.7	1.1	0.5	$UB1$	49.9
		EXM	1.8	2.5	1.0	UB	40.9
	$\mathcal{D}_2$	EXM-0	0.7	1.0	0.5	$UB1$	56.0
		EXM	1.8	2.4	0.9	UB	42.3
	$\mathcal{D}_3$	EXM-0	0.7	1.0	0.4	$UB1$	58.3
		EXM	1.7	2.2	0.9	UB	42.0
	$\mathcal{D}_4$	EXM-0	0.5	0.7	0.4	$UB1$	60.1
		EXM	1.5	1.8	0.8	UB	47.2

7. 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). We proved the importance of incorporating holding cost into the CTSNDP by showing that, in the best situation, the cost saving percentage of the minimal total cost can be arbitrarily large.

The exact algorithm uses the same dynamic discretization discovery (DDD) solution framework proposed by Boland et al. (2017) for the CTSNDP, but it extends it in a number of non-trivial ways by exploiting a new relaxation of a complete time-index model 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, in particular when the network is characterized by high flexibility and low connectivity levels. It was particularly note that, in average case, the cost saving percentage of the minimal total cost can be up to 40.8%.

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 a large number of 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).

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)'.
URL: *http://www.gurobi.com*

Hall, R. W. (1987), Consolidation strategy: inventory, vehicles and terminals, *Journal of Business Logistics* **8**(2), 57.

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

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.

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.

Needham, P. M. and Evers, P. T. (1998), The influence of individual cost factors on the use of emergency transshipments, *Transportation Research Part E: Logistics and Transportation Review* **34**(2), 149–160.

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.

Analysis of holding costs, glossary, and proofs of statements

EC.1. Best-case analysis of cost saving when incorporating holding cost

To highlight the importance of considering holding costs, Figure EC.1 shows a simple example involving four terminals, with two commodities in $\mathcal{K} = \{1, 2\}$ having their origins $o^1 = a$ and $o^2 = c$ and their destinations $d^1 = d^2 = d$. In the example, we assume that the commodities' earliest available times e^1 and e^2 are such that commodity 1 of demand q_1 can potentially wait t units of times to be consolidated with commodity 2 at terminal b at an additional holding cost $h_b^1 q_1 t$, modeled by the holding arc (b', b).

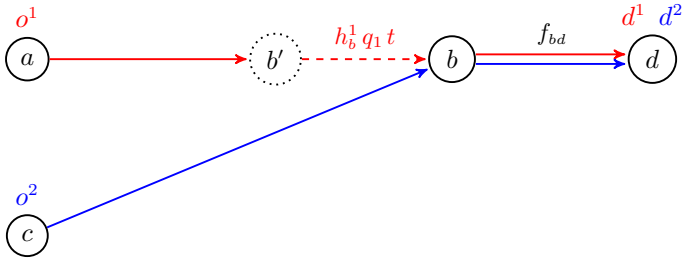


Figure EC.1 Best-case analysis of cost saving for incorporating holding cost

If the holding cost at terminal b is ignored when determining the optimal solution, the cost of the last leg $b-d$ without the flow costs is equal to $h_b^1 q_1 t + f_{bd}$ (i.e., the sum of the holding cost and fixed cost), where commodity 1 is consolidated at terminal b with commodity 2 after having waited for t units of time, and the two commodities are then routed together to terminal d. If the holding cost at terminal b is considered and $h_b^1 q_1 t > f_{bd}$, i.e., waiting at terminal b for commodity 1 incurs a holding cost greater than the fixed cost associated with the final arc (b, d), then in the corresponding optimal solution the two commodities are routed on two separate paths, namely (a, b, d) and (c, b, d), with a total cost of the corresponding final leg $b-d$ equal to $2f_{bd}$. Therefore, a total saving of $h_b^1 q_1 t - f_{bd}$ is achieved in the latter case, and the corresponding percentage $(h_b^1 q_1 t - f_{bd})/(2f_{bd}) \times 100\%$ computed with respect to the total cost of the optimal solution can be arbitrarily large when $h_b^1 q_1 t$ grows to infinity and f_{bd} is fixed.

EC.2. Glossary

Tables EC.1, EC.2 and EC.3 summarize the terminology and notation used in the paper.

Table EC.1 Glossary of symbols used: Problem description

Symbol	Meaning
$\mathcal{D}$	(flat) network $\mathcal{D}=(\mathcal{N},\mathcal{A})$
$\mathcal{N}$	node set of network $\mathcal{D}$
$\mathcal{A}$	arc set of network $\mathcal{D}$
$\mathcal{K}$	set of commodities
o^k	origin of commodity $k\in\mathcal{K}$
d^k	destination of commodity $k\in\mathcal{K}$
q^k	demand of commodity $k\in\mathcal{K}$
τ_{ij}	travel time of arc $(i,j)\in\mathcal{A}$
c_{ij}^k	per-unit-of-flow cost of arc $(i,j)\in\mathcal{A}$ and commodity $k\in\mathcal{K}$
f_{ij}	fixed cost of arc $(i,j)\in\mathcal{A}$
u_{ij}	capacity of arc $(i,j)\in\mathcal{A}$
e^k	earliest available time of commodity $k\in\mathcal{K}$
l^k	latest arrival time of commodity $k\in\mathcal{K}$
h_i^k	per-unit-of-demand-and-time (in-storage holding) cost of commodity $k\in\mathcal{K}$ and node $i\in\mathcal{N}$

Table EC.2 Glossary of symbols used: Solution representation and TI model

Symbol	Meaning
$P^k=(v_1^k,v_2^k,...,v_{\eta^k+1}^k)$	path for commodity $k\in\mathcal{K}$
$\mathcal{W}^k=(P^k,t^k)$	k-feasible timed path
$\mathcal{W}=\{\mathcal{W}^k\}_{k\in\mathcal{K}}$	feasible CTSNDP-HC solution
δ_n^k	waiting time at node n of P^k
$\mathcal{C}=\{\mathcal{C}_1,\mathcal{C}_2,...,\mathcal{C}_{\lvert\mathcal{C}\rvert}\}$, $\mathcal{C}_r=(\alpha_r,J_r)$, $r=1,2,...,\lvert\mathcal{C}\rvert$	set of consolidation plans
$\mathcal{P}=\{P^k\}_{k\in\mathcal{K}}$	a routing plan
$\mathcal{S}=(\mathcal{P},\mathcal{C})$	flat solution
$z_{fc}(\mathcal{S})$	cost (fixed and flow costs) of flat solution $\mathcal{S}$
$\mathcal{D}_\mathcal{T}^\Delta=(\mathcal{N}_\mathcal{T}^\Delta,\mathcal{H}_\mathcal{T}^\Delta\bigcup\mathcal{A}_\mathcal{T}^\Delta)$	time-expanded network with discretization level Δ
$\mathcal{T}_i$	set of time points of i and $\mathcal{T}=\bigcup_{i\in\mathcal{N}}\mathcal{T}_i$
SND-HC($\mathcal{D}_\mathcal{T}^\Delta$)	TI formulation for the CTSNDP-HC based on graph $\mathcal{D}_\mathcal{T}^\Delta$
$y_{ij}^{t\bar{t}}$	service decision variables
$x_{ij}^{kt\bar{t}}$	routing decision variables
w_i^k	holding time decision variables
IM($\mathcal{S}$)	implementable model
$z_w(\mathcal{S})$	optimal holding cost of flat solution $\mathcal{S}$

EC.3. Proof of statements

EC.3.1. 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).□

Table EC.3 Glossary of symbols used: Exact algorithm and relaxation

Symbol	Meaning
$\hat{\Delta}$	discretization associated with a complete TI model
$\mathcal{D}_{\hat{\mathcal{T}}}^{\hat{\Delta}}$	fully time-expanded network
$\mathcal{D}_{\mathcal{T}} = (\mathcal{N}_{\mathcal{T}}, \mathcal{H}_{\mathcal{T}} \bigcup \mathcal{A}_{\mathcal{T}})$	partially time-expanded network
$\rho_i(t)$, $\mu(a)$, $\sigma(a)$	mapping functions
$T(P^k)$	transit time of path P^k
$\xi_i(t)$	mapping function
SND-HC-R($\mathcal{D}_{\mathcal{T}}$)	relaxation of SND-HC($\mathcal{D}_{\hat{\mathcal{T}}}^{\hat{\Delta}}$)
LB, UB	lower and upper bound at a generic iteration of the exact algorithm
optimality tolerance	optimality tolerance parameter
$\mathcal{J}_C$, $\mathcal{J}_P$	index sets of infeasible consolidation constraints (12d)

EC.3.2. Proof of Theorem 1

Let $(\bar{x}, \bar{y}, \bar{w})$ be an optimal solution of formulation SND-HC($\mathcal{D}_{\hat{\mathcal{T}}}^{\hat{\Delta}}$) (i.e., an optimal CTSNDP-HC solution) of cost $\bar{z}$, and let $\overline{\mathcal{A}} = \{((i,t),(j,t+\tau_{ij})) \in \mathcal{A}_{\hat{\mathcal{T}}}^{\hat{\Delta}} : \bar{y}_{ij}^{t,t+\tau_{ij}} > 0\}$ 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 (18) and (19), 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

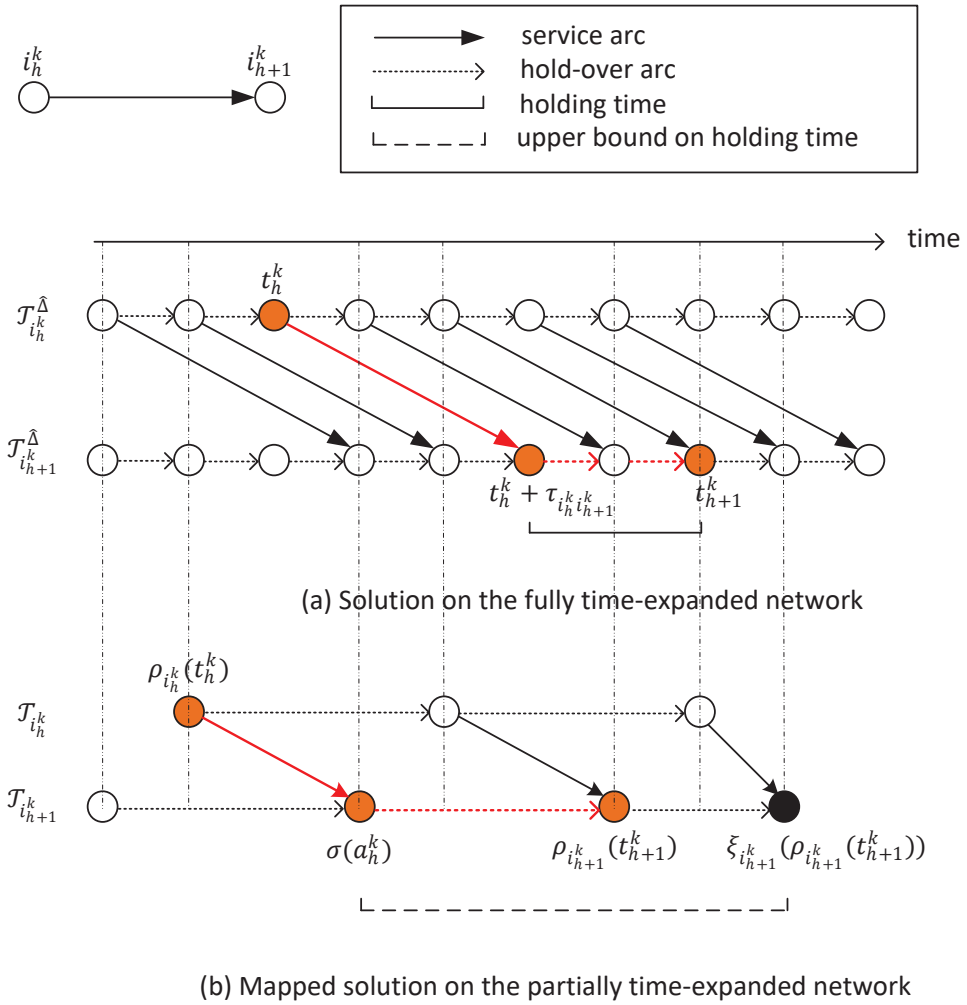


Figure EC.2 Illustration of the mapping functions $\rho(.)$, $\sigma(.)$ and $\xi(i)$

total fixed, flow and holding cost. For each path P^k, $k \in \mathcal{K}$, we have that equation (19) follows from expression (EC.1) and (EC.2). Constraints (18) for the values w_i^k and path P^k are as follows:

$$w_i^k \leq \begin{cases} \xi_{i_1^k}(\rho_{i_1^k}(t_1^k)) - e^k, & i = o^k, \\ l^k - \sigma(a_{\eta^k}^k), & i = d^k, \\ \xi_{i_h^k}(\rho_{i_h^k}(t_h^k)) - \sigma(a_{h-1}^k), & i = i_h^k, h = 2, ..., \eta^k, \\ 0, & \text{otherwise}, \end{cases} \quad \forall\, i \in \mathcal{N}. \tag{EC.3}$$

Based on the mapping functions $\rho(.)$, $\sigma(.)$ and $\xi(i)$ (see Figure EC.2), we have that:

(i) $\xi_{i_h^k}(\rho_{i_h^k}(t_h^k)) \geq t_h^k$, for $h = 1, \ldots, \eta^k$, and $\xi_{i_1^k}(\rho_{i_1^k}(t_1^k)) - e^k \geq t_1^k - e^k = \overline{w}_{o^k}^k$.

(ii) $\sigma(a_{h-1}^k) \leq t_{h-1}^k + \tau_{i_{h-1}^k i_h^k}$ $h = 2, \ldots, \eta^k$, and $l^k - \sigma(a_{\eta^k}^k) \geq l^k - (t_{\eta^k}^k + \tau_{i_{\eta^k}^k d^k}) = \overline{w}_{d^k}^k$.

(iii) $\xi_{i_h^k}(\rho_{i_h^k}(t_h^k)) - \sigma(a_{h-1}^k) \geq t_h^k - (t_{h-1}^k + \tau_{i_{h-1}^k i_h^k}) = \overline{w}_{i_h^k}^k$, $h = 2, ..., \eta^k$.

(iv) For each $i \notin P^k$, we have $w_i^k \leq 0$, 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.3.3. 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 $x_{ij}^{k\bar{t}\bar{t}}=\overline{x}_{ij}^{k\bar{t}\bar{t}}$, $k\in\mathcal{K}$, and $y_{ij}^{\bar{t}\bar{t}}=\overline{y}_{ij}^{\bar{t}\bar{t}}$ for all arcs in $((i,\bar{t}),(j,\bar{t}))\in(\mathcal{H}_{\mathcal{T}}\cup\mathcal{A}_{\mathcal{T}})\cap(\overline{\mathcal{H}}_{\mathcal{T}}\cup\overline{\mathcal{A}}_{\mathcal{T}})$. 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 $((i,t),(j,t'))\in\overline{A}_{\mathcal{T}}$ with $t''<t'$ for all $((i,t),(j,t''))\in A_{\mathcal{T}}$. For each commodity $k\in\mathcal{K}$, let $\overline{P}^k=(a_1^k,\ldots,a_{\eta^k}^k)$, $a_h^k\in\overline{\mathcal{A}}$, $h=1,\ldots,\eta^k$, be the path induced by solution $(\bar{x},\bar{y})$ with $a_h^k=((i_h^k,\bar{t}_h^k),(i_{h+1}^k,\overline{\pi}_{h+1}^k))$, $h=1,\ldots,\eta^k$, where $\bar{t}_h^k$ is the departure time at node i_h^k and $\overline{\pi}_h^k$ is the corresponding arrival time, $h=1,\ldots,\eta^k+1$. 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 $P^k=\overline{P}^k=(a_1^k,\ldots,a_{\eta^k}^k)$, $a_h^k\in\overline{\mathcal{A}}$, with departure time $t_h^k=\bar{t}_h^k$ and arrival times $\pi_h^k\le\overline{\pi}_h^k$, $h=1,\ldots,\eta^k+1$.

We now show that solution vector $w=\overline{w}$ satisfies constraints (18) and (19) 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 (19) are satisfied. Define $\overline{N}_k=\bigcup_{h=1,\ldots,\eta^k+1}\{i_h^k\}$, $k\in\mathcal{K}$, as the set of nodes visited by path $\overline{P}_k$. Then, we have that solution vector $\overline{w}$ is defined as:

$$\overline{w}_i^k\le\begin{cases}\xi_i(\bar{t}_1^k)-e^k, & i=o^k,\\ l^k-\overline{\pi}_{i_{\eta^k}^k}^k, & i=d^k,\\ \xi_i(\bar{t}_h^k)-\overline{\pi}_h^k, & \text{if } i=i_h^k\in\overline{N}_k\setminus\{o^k,d^k\},\\ 0, & \text{otherwise,}\end{cases}\qquad\forall\ i\in\mathcal{N}. \tag{EC.4}$$

and for inequalities (18) we have

$$w_i^k\le\begin{cases}\xi_i(t_1^k)-e^k=\xi_i(\bar{t}_1^k)-e^k, & i=o^k,\\ l^k-\pi_{i_{\eta^k}^k}^k\ge l^k-\overline{\pi}_{i_{\eta^k}^k}^k, & i=d^k,\\ \xi_i(t_h^k)-\pi_h^k\ge\xi_i(\bar{t}_h^k)-\overline{\pi}_h^k, & \text{if } i=i_h^k\in\overline{N}_k\setminus\{o^k,d^k\},\\ 0, & \text{otherwise,}\end{cases}\qquad\forall\ i\in\mathcal{N}. \tag{EC.5}$$

Hence, $w=\overline{w}$ is proved to be a feasible solution for inequalities (18).□