HAL open science

New Formulations for the Scheduled Service Network Design Problem

Mike Hewitt, Fabien Lehuédé

To cite this version:

Mike Hewitt, Fabien Lehuédé. New Formulations for the Scheduled Service Network Design Problem. Transportation Research Part B: Methodological, 2023, 172, pp.117-133. 10.1016/j.trb.2023.04.002 . hal-03797949v2

HAL Id: hal-03797949

https://hal.science/hal-03797949v2

Submitted on 17 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HAL
open science

New Formulations for the Scheduled Service Network Design Problem

Mike Hewitt, Fabien Lehuédé

To cite this version:

Mike Hewitt, Fabien Lehuédé. New Formulations for the Scheduled Service Network Design Problem. 2022. hal-03797949

HAL Id: hal-03797949

https://hal.science/hal-03797949

Preprint submitted on 5 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

New Formulations for the Scheduled Service Network Design Problem

Mike Hewitt[a,b], Fabien Lehuédé[b]

[a] *Department of Information Systems and Supply Chain Management Quinlan School of Business Loyola University Chicago*
[b] *IMT Atlantique LS2N UMR CNRS Nantes France*

Abstract

We propose a new approach to formulating the Scheduled Service Network Design Problem (SSNDP) that involves modeling with enumerated consolidations of shipments routed on the physical network. This is in contrast to the classical approach of capturing the synchronization of vehicles and shipments needed for consolidation with a time-expanded network. The proposed formulation has both a stronger linear relaxation and is less symmetric. We present multiple speed-up techniques and with an extensive computational study illustrate that the consolidation-based formulation is much easier to solve with an off-the-shelf solver than the classical formulation based on a time-expanded network.

To avoid instances involving large numbers of consolidations we propose a hybrid formulation that combines ideas from the consolidation and time-expanded network-based approaches to formulating the SSNDP. We show that instances of the hybrid formulation are much easier to solve than both instances of the pure consolidation-based formulation and those based on a time-expanded network formulation. Finally, we discuss how formulating with consolidations facilitates modeling issues that have not yet been addressed in the literature, such as bin-packing considerations when computing vehicle capacity needs. In addition, the proposed modeling technique for bin-packing considerations in a consolidation-based formulation yields instances that are easier to solve than those wherein capacity is modeled in an aggregate sense.

Keywords: Freight Transportation, Scheduled Service Network Design Problem, Integer Programming

1. Introduction

The Scheduled Service Network Design Problem (SSNDP) is a classic optimization problem that can be used to plan the inter-city transportation operations of carriers that specialize in shipments that are small relative to vehicle capacity (Crainic 2000, Wieberneit 2008). Examples of such carriers can be found in multiple industries that generate billions of USD a year in revenues world-wide, including parcel and small package as well as Less-than-truckload (LTL) freight. Both of these industries are critical to economies around the world, with LTL carriers supporting many supply chains and parcel and small package carriers supporting eCommerce sales. In the United States the LTL industry was reported to be $46 billion in 2021 (Schulz 2021) while UPS reported $69.44 billion in revenue from its US and international small package operations in 2020 (UPS 2021). For every mode of transportation other than pipeline, one can find examples of carriers for whom the SSNDP is relevant (Barnhart and Schneur 1996, Bakir, Erera, and Savelsbergh 2021, Chouman and Crainic 2021, Christiansen et al. 2021). Given the markets they serve, a key to the profitability of these carriers is the ability to consolidate multiple shipments into the same vehicle dispatch, increasing vehicle utilization and decreasing transportation costs.

To enable consolidation, such carriers do not route shipments directly from customer origin to destination. Instead, they are routed through a network of consolidation terminals at which shipments can be transferred from inbound vehicles to outbound vehicles. More precisely, they are routed on a path through the terminal network that begins at its origin terminal, visits one or more intermediate terminals, and ends at its destination terminal. Typically, each shipment is to be picked up at its origin terminal no earlier than the time at which it is expected to be available and delivered to its destination terminal no later than the time at which it is due. A shipment need not move continuously through the terminal network. Instead, it can be held at both the origin and intermediate terminals on its path. Transporting shipments between terminals requires the use of one or more vehicles, each of which has a capacity, potentially along multiple dimensions. Transportation costs are incurred on a per-vehicle basis that depends on distances traveled.

Many variants of the SSNDP have been studied in the literature. However, the classical SSNDP seeks to determine a path for each shipment through the terminal network and a dispatch time for each terminal to terminal movement on this path. It also determines the number of vehicles and their corresponding dispatch times on each terminal to terminal movement within the network. Constraining these decisions are the need to respect shipment available and due times, travel times, and to ensure that sufficient vehicle capacity is dispatched on each movement to carry the shipments to be transported. The objective the SSNDP considers is to minimize total vehicle transportation costs. A recent survey on Service Network Design, including a description of the SSNDP is provided in Crainic and Hewitt (2021).

Consolidation is possible when multiple shipments dispatch on the same terminal to terminal movement at the same time. The classical integer programming formulation of the SSNDP captures this synchronization by modeling the routing of shipments on a directed *time-expanded network* consisting of nodes and arcs. In such a network, activities at a terminal at different points in time are modeled with different nodes. Similarly, a terminal to terminal movement dispatched at different times is modeled with different arcs. As such, the opportunity for consolidation is recognized in this formulation when multiple shipments travel on the same arc in the time-expanded network. Capacity needs are captured by ensuring a sufficient number of vehicles also travel on that same arc with knapsack-type linking constraints.

Time-expanded network-based formulations of the SSNDP have received significant attention in the academic literature, and from various perspectives. Many papers have addressed extending the scope of decision-making considered by the SSNDP to include additional management issues. A classic and important example is the need to move vehicles empty to position them for future transportation moves. This is sometimes referred to as *empty repositioning*, or, more generally, *asset* or *resource management* (Hewitt et al. 2019, Scherr et al. 2019, Crainic et al. 2016, Erera et al. 2013, Andersen et al. 2011, Teypaz, Schrenk, and Cung 2010, Jarrah, Johnson, and Neubert 2009, Andersen, Crainic, and Christiansen 2009). Other papers have focused on adapting and extending the SSNDP to issues particular to certain modes of transportation. One example is rail, wherein consolidation can happen at multiple levels (Chouman and Crainic 2021, Zhu, Crainic, and Gendreau 2014). Many papers have focused on algorithmic techniques for the SSNDP, with some proposing exact methods (Crainic and Gendron 2021, Marshall et al. 2020, Hewitt 2019, Boland et al. 2017) and others heuristics (Crainic and Gendreau 2021, Chouman and Crainic 2015). We note that in this paper we focus on solving the SSNDP exactly. Lastly, regarding how time is discretized we are only aware of Boland et al. (2019), which observes that the finer the granularity of the discretization used to construct the time-expanded network, the higher the quality of solutions to the resulting instance of the SSNDP.

Using a time-expanded network-based formulation leads to multiple computational chal-

lenges when solving instances of the SSNDP. The first challenge is that the level of discretization of time required to find high-quality solutions can lead to extremely large time-expanded networks. This in turn leads to large integer programs that are difficult to solve. This challenge is somewhat mitigated by the *Dynamic Discretization Discovery* (DDD) framework presented in Boland et al. (2017) and further studied in Hewitt (2019), Marshall et al. (2020), which propose generating such networks iteratively as opposed to in a static, *a priori* manner. While DDD has been shown to be computationally effective, it may be challenging to implement.

The duplication of physical links for different points in time also leads to the second challenge, which is that instances of the SSNDP formulated on a time-expanded network tend to be very symmetric. As most integer programming solvers rely on a variable branching-type mechanism to search the space of solutions, much of this time can be wasted. While research has been done to reduce symmetry in mixed integer programs in general (Pfetsch and Rehn 2019, Margot 2010), we are unaware of any research that focuses on symmetries enabled by formulating on a time-expanded network.

A third challenge related to solving time-expanded network-based formulations is that the knapsack-type linking constraints used to model capacity needs lead to notoriously weak linear programming relaxations. As most integer programming solvers rely on bounding techniques that involve solving linear programming relaxations to establish the quality of a solution, the time needed to establish that a solution is optimal (within a given tolerance) can be long. This challenge is somewhat mitigated by *Flow cover inequalities* (Nemhauser and Wolsey 1988), which are implemented in many commercial integer programming solvers. While these three challenges are well-known, to the best of our knowledge, all research on the SSNDP to date has relied on a time-expanded network-based formulation.

In this paper we propose new formulations for the SSNDP with the ultimate goal that these formulations can be effectively solved by an off-the-shelf mixed integer programming (MIP) solver. Our work is based on the concept of enumerated consolidations, introduced in Hewitt and Lehuédé (2022) for a special case of the SSNDP in which the path of each commodity is known before solving the problem.

The contributions of this paper are the following:

- We propose a new formulation for the SSNDP. This formulation does not involve a time-expanded network and instead prescribes decisions at the level of the physical terminal network. To recognize shipment synchronization needs required by consolidation, the formulation is based on explicit enumerations of consolidations.

- We propose two enhancements to make the solving instances of the formulation more efficient. First, we propose valid inequalities to further strengthen the formulation. Second, we propose a pruning mechanism to recognize consolidations that need not be modeled and reduce symmetries in the model.

- We connect the new formulation to the classical time expanded network formulation. This results in a hybrid formulation that is able to capture desirable properties of both "pure" formulations and can be adapted depending on data.

- We show that the consolidation based formulation can model operational issues that are challenging to represent with a time-expanded formulation. Namely, bin packing-type constraints and piecewise linear transportation costs. In a SSNDP context, bin packing constraint model the fact that a commodity cannot be split over several vehicles. To our knowledge, this is the first time this important feature is modeled in the SSNDP. Piecewise linear costs can occur when cost parameters (e.g. per-unit-of-distance costs) depend on the amount transported.

In an extensive set of computational experiments we find that:

- An optimization solver can solve more instances of the consolidation-based formulation than those based on the classical time-expanded network-based formulation. This is explained by showing that the consolidation based formulation has a stronger linear relaxation.
- The additional enhancements (pruning mechanism, valid inequalities and hybrid model) enable significantly more instances to be solved within the time limit.
- Regarding modeling, we show that incorporating empty repositioning considerations into instances of the consolidation-based formulations does not make them significantly harder to solve. Similarly, the proposed procedure for incorporating bin-packing considerations leaves instances that are actually easier to solve than when capacity is modeled in an aggregate sense.
- Finally, we show that ignoring bin-packing constraints leads to a significant under-estimate of costs, and that under-estimate increases as the number of shipments to transport gets larger.

The rest of this paper is structured as follows. Section 2 grounds the paper by presenting a mathematical notation relevant to the formulations of the SSNDP considered in this paper as well as the classical time-expanded network-based formulation. Section 3 presents the pure consolidation-based formulation of the SSNDP along with techniques for strengthening that formulation. Section 4 presents the hybrid formulation while section 5 proposes modifications to the consolidation-based formulations to capture issues such as empty repositioning and bin packing. Section 6 describes the computational study of the proposed formulations and presents findings. Finally, section 7 concludes the paper and presents avenues for future research.

2. Mathematical notation and time-expanded network formulation

To ground the paper, in this section we present the classical time-expanded network formulation of the SSNDP. We also present notation that is useful both for that formulation and the new ones proposed later.

Regarding notation, we presume a directed network $\mathcal{D} = (\mathcal{N}, \mathcal{A})$, where the set $\mathcal{N}$ models consolidation terminals and the set $\mathcal{A}$ models physical transportation moves between terminals. Regarding transportation, there is a fixed cost of f_{ij} incurred for each dispatch of a vehicle of capacity u_{ij} on arc $(i, j) \in \mathcal{A}$. We presume the travel time on arc $(i, j) \in \mathcal{A}$ is τ_{ij} and is independent of the departure time. There is a set of shipments $\mathcal{K}$. Associated with each shipment $k \in \mathcal{K}$ is a terminal where it is to be picked up, o_k, no earlier than the release date e_k, and is to be delivered, d_k, no later than the due date l_k. In addition, associated with shipment k is the size q_k of the shipment, expressed in the same unit as vehicle capacity.

Regarding the routing of shipments, we presume for each shipment $k \in \mathcal{K}$ there is a known set of paths P_k from o_k to d_k, that model potential sequences of movement(s) for that shipment from origin to destination. A single such path must be chosen for each shipment. As further notation, we let the set $\mathcal{K}_{ij}$ denote the set of shipments k for which there is a path $p \in P_k$ that contains the arc (i, j). Relatedly, we let $P(i, j)_k \subseteq P_k$ denote the set of paths in P_k that contain arc (i, j). We also presume that for each shipment and each movement in its chosen path the entire shipment must dispatch at the same time. Lastly, we note that for each shipment $k \in \mathcal{K}$, path $p \in P_k$, and arc $(i, j) \in p$ we can derive a time window $[\alpha_{ij}^{kp}, \beta_{ij}^{kp}]$ for when shipment k can dispatch on arc (i, j) if it takes path p.

The classical time-expanded network formulation of the SSNDP assumes a discretization of time, $\mathcal{T} = \{1, \ldots, T\}$, of a planning horizon that is T periods long. This discretization of time is used to construct the time-expanded network $\mathcal{D}_\mathcal{T} = (\mathcal{N}_\mathcal{T}, \mathcal{A}_\mathcal{T})$. The node set $\mathcal{N}_\mathcal{T}$ consists of nodes of the form $(i,t), i \in \mathcal{N}, t \in \mathcal{T}$, that model actions that occur at terminal i during the time period represented by t. The set $\mathcal{A}_\mathcal{T}$ consists of two sets of arcs. The first set contains arcs of the form $((i,t),(j,t')), (i,t),(j,t') \in \mathcal{N}_\mathcal{T}, i \neq j, t' = \min(t'' \in \tau | t'' - t \geq \tau_{ij})$ that model traveling from terminal i at time t to arrive at terminal j at time t'. The second set contains arcs of the form $((i,t),(i,t+1)), (i,t),(i,t+1) \in \mathcal{N}_\mathcal{T}$ that model idling at terminal i from period t to period $t+1$.

On this network, we define the integer variables $y_{ij}^{tt'}, ((i,t),(j,t')) \in \mathcal{A}_\mathcal{T}$ to represent the number of vehicles to be dispatched on arc $((i,t),(j,t'))$. For a shipment k, the choice of a path $p \in \mathcal{P}_k$ is identified by a variable v_p^k which is equal to one if shipment k follows path p. To model the routing of shipments on arcs we let the binary variables $x_{ij}^{ktt'}, k \in \mathcal{K}, ((i,t),(j,t')) \in \mathcal{A}_\mathcal{T}$ denote whether shipment k takes arc $((i,t),(j,t'))$. Given these decision variables, the SSNDP can be formulated as follows. We label this formulation $TEN(\mathcal{D})$.

$$z_{TEN(\mathcal{D})} = \text{minimize} \sum_{((i,t),(j,t')) \in \mathcal{A}_\mathcal{T}} f_{ij} y_{ij}^{tt'}$$

subject to

$$\sum_{p \in P_k} v_p^k = 1 \qquad \forall k \in \mathcal{K}, \tag{1}$$

$$\sum_{p \in P(i,j)_k} v_p^k = \sum_{(t,t'):((i,t),(j,t')) \in \mathcal{A}_\mathcal{T}} x_{ij}^{ktt'} \qquad \forall (i,j) \in T, k \in \mathcal{K}_{ij}, \tag{2}$$

$$\sum_{((i,t),(j,t')) \in \mathcal{A}_\mathcal{T}} x_{ij}^{ktt'} - \sum_{((j,t'),(i,t)) \in \mathcal{A}_\mathcal{T}} x_{ji}^{kt't} = \begin{cases} 1 & (i,t) = (o_k, e_k) \\ -1 & (i,t) = (d_k, l_k) \\ 0 & \text{o.w.} \end{cases} \qquad \forall (i,t) \in \mathcal{N}_\mathcal{T}, \forall k \in \mathcal{K}, \tag{3}$$

$$\sum_{k \in \mathcal{K}} q_k x_{ij}^{ktt'} \leq u_{ij} y_{ij}^{tt'} \qquad \forall ((i,t),(j,t')) \in \mathcal{A}_\mathcal{T}, \tag{4}$$

$$y_{ij}^{tt'} \in \mathbb{N} \qquad \forall ((i,t),(j,t')) \in \mathcal{A}_\mathcal{T}, \tag{5}$$

$$v_p^k \in \{0,1\} \qquad \forall k \in \mathcal{K}, p \in P(i,j)_k, \tag{6}$$

$$x_{ij}^{ktt'} \in \{0,1\} \qquad \forall ((i,t),(j,t')) \in \mathcal{A}_\mathcal{T}, k \in \mathcal{K}. \tag{7}$$

The objective function seeks to minimize the sum of transportation cost. Constraints (1) state that each commodity is assigned a path. Constraints (2) connect the commodity arc and path variables. Constraints (3) are classical flow balance constraints, albeit expressed on the time-expanded network. Constraints (4) ensure sufficient vehicle capacity on timed arcs. Constraints (5)-(7) define the decision variables and their domains. There are standard pre-processing techniques for removing unnecessary variables from an instance of the $TEN(\mathcal{D})$ formulation. One example is to only create the variable $y_{ij}^{tt'}$ if $\exists k \in \mathcal{K}$ such that $x_{ij}^{ktt'}$ is created.

3. Consolidation-based formulation

We next present a consolidation-based formulation of the SSNDP. There are two significant differences between this formulation and $TEN(\mathcal{D})$. The first is that it is not defined on a time-expanded network like $\mathcal{D}_\mathcal{T}$ but instead the network $(\mathcal{N}, \mathcal{A})$. The second is that it is based on an

a priori enumeration of consolidations on each arc $(i,j) \in \mathcal{A}$, wherein a *consolidation* is defined as a set of shipments that dispatch on the same physical move at the same time.

Regarding notation for this formulation, we let Ω_{ij} define the set of potential shipment consolidations on arc (i,j). Thus, an element $\omega \in \Omega_{ij}$ is a subset of shipments (i.e. $\omega \subseteq \mathcal{K}$). For $(i,j) \in \mathcal{A}, \omega \in \Omega_{ij}$, we let ϕ_ω^k indicate whether shipment $k \in \mathcal{K}$ is contained in consolidation ω. Relatedly, we let s_ω define the number of vehicles needed to transport consolidation $\omega \in \Omega_{ij}$. This formulation can mimic how capacity needs are modeled with $TEN(\mathcal{D})$ by setting $s_\omega = \lceil \frac{\sum_{k \in \omega} q_k}{u_{ij}} \rceil$. However, the values s_ω can be used to model more accurate or complex capacity considerations. For example, it is common for the capacity of a vehicle to involve multiple dimensions, such as weight and cubic volume. Capturing multiple dimensions of capacity in the $CONS(\mathcal{D})$ simply requires accurate computation of vehicle needs, s_ω. We present other examples later in the paper.

Given this notation, we define the binary decision variable δ_ω to indicate whether the consolidation $\omega \in \Omega_{ij}$ is chosen. The integer decision variable z_{ij} is used to model the number of vehicles that dispatch on arc $(i,j) \in \mathcal{A}$. Like the formulation $TEN(\mathcal{D})$, the choice of physical routing of shipments is done with the binary variables $v_p^k \in \{0,1\}, p \in P_k, k \in \mathcal{K}$. The decision variable γ_{ij}^k represents the time at which shipment k dispatches on arc $(i,j) \in \mathcal{A}$. The consolidation-based formulation we present, which we label $CONS(\mathcal{D})$, is as follows.

$$z_{CONS(\mathcal{D})} = \text{minimize} \sum_{(i,j)\in\mathcal{A}} f_{ij} z_{ij}$$

subject to

(1)

$$\sum_{\omega\in\Omega_{ij}} \phi^k_\omega \delta_\omega = \sum_{p\in P(i,j)_k} v^k_p \qquad \forall (i,j)\in\mathcal{A}, k\in\mathcal{K}_{ij}, \tag{8}$$

$$\sum_{\omega\in\Omega_{ij}} s_\omega \delta_\omega \le z_{ij} \qquad \forall (i,j)\in\mathcal{A}, \tag{9}$$

$$\gamma^k_{ij} - \gamma^{k'}_{ij} \le T(1 - \sum_{\omega\in\Omega_{ij}} \phi^k_\omega \phi^{k'}_\omega \delta_\omega) \qquad \forall (i,j)\in\mathcal{A}, k,k'\in\mathcal{K}_{ij}, \tag{10}$$

$$\sum_{p\in P(i,j)_k} \alpha^{kp}_{ij} v^k_p \le \gamma^k_{ij} \qquad \forall (i,j)\in\mathcal{A}, k\in\mathcal{K}_{ij}, \tag{11}$$

$$\gamma^k_{ij} \le \sum_{p\in P(i,j)_k} \beta^{kp}_{ij} v^k_p \qquad \forall (i,j)\in\mathcal{A}, k\in\mathcal{K}_{ij}, \tag{12}$$

$$\sum_{(o_k,j)\in\mathcal{A}} \gamma^k_{o_k j} \ge e_k \qquad \forall k\in\mathcal{K}, \tag{13}$$

$$\sum_{(i,d_k)\in\mathcal{A}} (\gamma^k_{id_k} + \tau_{id_k}(\sum_{p\in P(i,d_k)_k} v^k_p)) \le l_k \qquad \forall k\in\mathcal{K}, \tag{14}$$

$$\sum_{(i,j)\in\mathcal{A}} (\gamma^k_{ij} + \tau_{ij}(\sum_{p\in P(i,j)_k} v^k_p)) \le \sum_{(j,i)\in\mathcal{A}} \gamma^k_{ji} \qquad \forall j\in N, k\in\mathcal{K}, \tag{15}$$

$$\gamma^k_{ij} \ge 0 \qquad \forall k\in\mathcal{K}, (i,j)\in\mathcal{A},, \tag{16}$$

$$z_{ij} \in \mathbb{N} \qquad \forall (i,j)\in\mathcal{A}, \tag{17}$$

$$\delta_\omega \in \{0,1\} \qquad \forall \omega\in\Omega_{ij}, (i,j)\in\mathcal{A}, \tag{18}$$

(6).

Constraints (8) ensure that for each arc in the path chosen for a shipment, a consolidation is chosen that contains that shipment. Constraints (9) ensure sufficient vehicles are dispatched on each physical transportation move given the consolidations chosen for that move. Constraints (11) and (12) ensure if the shipment travels on a path that contains an arc then the dispatch time falls within the corresponding time window. Constraints (12) also ensure that for a given arc and shipment if a path containing that arc is not chosen for that shipment then the dispatch time is zero. Constraints (10) ensure that all pairs of shipments in a chosen consolidation for a physical transportation move dispatch at the same time. Constraints (13) ensure each shipment dispatches from its origin at a time after its release date. Similarly, constraints (14) ensure each shipment does not arrive at its destination after its due date. Constraints (15) ensure that the continuous dispatch variables agree with transportation times. Finally, constraints (16),(17) and (18) define the additional decision variables needed for this formulation.

Hewitt and Lehuédé (2022) show analytically that their proposed consolidation-based formulation for the Service Network Scheduling Problem, the SSNDP wherein shipment paths have already been determined, is stronger than a time-expanded network formulation of the problem. That analysis can be easily repeated in the more general context of the SSNDP.We next present two techniques for reducing the time required to solve instances of *CONS*($\mathcal{D}$). The first is a technique for removing variables that introduce symmetry into the formulation while the second involves valid inequalities.

3.1. Consolidation Pruning

We first present a pruning rule for when a consolidation can be omitted from the *CONS*($\mathcal{D}$) formulation without rendering all optimal solutions infeasible. First, for a given arc $(i,j) \in \mathcal{A}$ consider a consolidation $\omega \subseteq \Omega_{ij}$ and the value s_ω. Next, consider a partition of ω into two subsets ω^1, ω^2 with values $s_{\omega^1}, s_{\omega^2}$ and suppose that $s_\omega = s_{\omega^1} + s_{\omega^2}$. We prove in Lemma 3.1 that in this case the consolidation ω can be removed from the formulation as any solution that selects consolidation ω can instead select consolidations ω^1 and ω^2.

THEOREM Consider a given arc $(i,j) \in \mathcal{A}$ and consolidation $\omega \subseteq \Omega_{ij}$ such that there exists $\omega^1, \omega^2 \subseteq \omega$ wherein $\omega^1 \cup \omega^2 = \omega, \omega^1 \cap \omega^2 = \emptyset$ and $s_\omega = s_{\omega^1} + s_{\omega^2}$. The optimal objective function value of the instance of *CONS*($\mathcal{D}$) formulated without ω is the same as the instance formulated with ω.

Proof. Consider a solution $(\bar{\gamma}, \bar{z}, \bar{\delta}, \bar{v})$ such that $\bar{\delta}_\omega = 1$. Note this implies that $\bar{z}_{ij} \geq s_\omega$. Note that because of constraints (10) we have $\bar{\gamma}^k_{ij} = \bar{\gamma}^{k'}_{ij} \quad \forall k, k' \in \omega$. Next, consider a solution $(\tilde{\gamma}, \tilde{z}, \tilde{\delta}, \tilde{v})$ wherein all decision variables take on the same values as in $(\bar{\gamma}, \bar{z}, \bar{\delta}, \bar{v})$, except $\tilde{\delta}_\omega = 0$ and $\tilde{\delta}_{\omega^1} = \tilde{\delta}_{\omega^2} = 1$. We note that as $s_\omega = s_{\omega^1} + s_{\omega^2}$ we have $\tilde{z}_{ij} = \bar{z}_{ij} \geq s_\omega = s_{\omega^1} + s_{\omega^2}$. Similarly, as $\omega^1, \omega^2 \in \omega$ we have $\tilde{\gamma}^k_{ij} = \tilde{\gamma}^{k'}_{ij} \quad \forall k, k' \in \omega^q, q = 1, 2$. Thus, the solution $(\tilde{\gamma}, \tilde{z}, \tilde{\delta}, \tilde{v})$ is feasible for *CONS*($\mathcal{D}$) and has same objective function value. □ □

We refer to the procedure of removing consolidations such as ω as *Consolidation Pruning (CP)*.

3.2. Valid inequalities

Regarding valid inequalities, we note that classical cutset-type inequalities for network design problems (Chouman, Crainic, and Gendron 2017) can be adapted to this formulation. For example, ensuring a sufficient number of vehicles depart a terminal to transport the shipments that originate at that terminal can be expressed as in constraints (19). To define these constraints we first compute $\underline{u_i} = min_{j:(i,j)\in A} u_{ij}$.

$$\sum_{j:(i,j)\in\mathcal{A}} z_{ij} \geq \lceil \frac{\sum_{k\in K:o_k=i} q_k}{\underline{u_i}} \rceil \quad \forall i \in \mathcal{N}. \tag{19}$$

The analogous inequality regarding shipments destined for terminal i can also be used to strengthen *CONS*($\mathcal{D}$). This type of inequality is denoted *Vehicle Cutset (VC)* inequalities in the following.

Similar reasoning enables the derivation of a new inequality. Namely, recognizing that for each shipment there must be a consolidation chosen for an arc that departs its origin we have the following valid inequality.

$$\sum_{j:(o_k,j)\in\mathcal{A}} \sum_{\omega\in\Omega_{o_k j}} \phi^k_\omega \delta_\omega = 1 \quad \forall k \in \mathcal{K}. \tag{20}$$

We call these inequalities *Origin Consolidation Cutset* (OCC) inequalities. There are analogous inequalities for each shipment and its destination terminal, which we refer to as *Destination Consolidation Cutset* (DCC) inequalities.

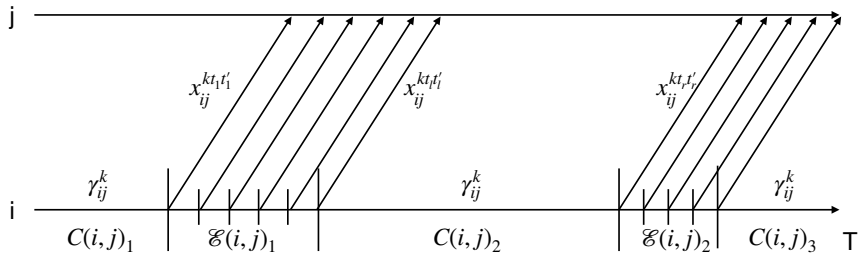

Figure 1: Modeling dispatch times on an arc differently in different partitions of the time horizon

4. Hybrid formulation

In this section, we present a hybrid formulation that combines elements of the consolidation-based formulation and the classical time-expanded network formulation. The primary motivation for this formulation is to avoid computational challenges associated with solving instances of $CONS(\mathcal{D})$ involving large numbers of consolidations, even after pruning. One such challenge is enumerating the consolidations. Another is formulating and solving the resulting mixed integer program as it will have a large number of binary decision variables.

Consider a single arc and interval of time. The number of consolidations associated with that arc and interval of time is exponential in the number of shipments that can dispatch on that arc during that interval. Thus, if the number of shipments is large, the number of consolidations will be as well. By instead modeling synchronization of shipment dispatches with timed copies of arcs, the number of consolidations can be greatly reduced. Thus, the hybrid formulation we propose relies on creating a partition of the time horizon for each arc. During some elements of the partition, shipment and vehicle movements on arcs are modeled as in the consolidation-based formulation. During other elements, they are modeled as in a time-expanded network.

More precisely, we presume that for each arc $(i,j) \in \mathcal{A}$ the planning horizon $\mathcal{T} = \{1,\ldots,T\}$ has been partitioned into time intervals $\mathcal{C}(i,j)_q, q = 1,\ldots,m_c$ and $\mathcal{E}(i,j)_q, q = 1,\ldots,m_e$. The intervals $\mathcal{C}(i,j)_q$ contain times during which a shipment dispatching on that arc is to be modeled with a continuous dispatch variable. The intervals $\mathcal{E}(i,j)_q$ contain times during which a shipment dispatching on that arc is to be modeled with a timed copy of that arc. In this formulation, the set $\mathcal{A}_\mathcal{T}$ contains only a subset of the arcs used to form $TEN(\mathcal{D})$. More precisely, the set $\mathcal{A}_\mathcal{T} = \{((i,t),(j,t')) : (i,j) \in \mathcal{A}, \exists q \| t \in \mathcal{E}(i,j)_q\}$. We note that for this formulation $\mathcal{A}_\mathcal{T}$ does not contain arcs of the form $((i,t),(i,t+1)), i \in \mathcal{N}$. We illustrate this partition of time and resulting modeling of dispatch times for a single arc in Figure 1.

Notationally-speaking, the formulation involves the same decision variables as presented above. The objective function combines the terms from the formulations presented above. Some constraint sets are the same as a set from one of the formulations presented above while others are a hybrid of constraints from each. We present the complete formulation below, even though some constraints have already been presented, for clarity.

$$\text{minimize} \sum_{(i,j)\in\mathcal{A}} f_{ij} z_{ij} + \sum_{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}} f_{ij} y_{ij}^{tt'}$$

subject to

$$\sum_{p\in\mathcal{P}_k} v_p^k = 1 \qquad \forall k \in \mathcal{K}, \tag{21}$$

$$\sum_{\omega\in\Omega_{ij}} s_\omega \delta_\omega \leq z_{ij} \qquad \forall (i,j) \in \mathcal{A}, \tag{22}$$

$$\gamma_{ij}^k - \gamma_{ij}^{k'} \leq T(1 - \sum_{\omega\in\Omega_{ij}} \phi_\omega^k \phi_\omega^{k'} \delta_\omega) \qquad \forall (i,j) \in \mathcal{A}, k, k' \in \mathcal{K}_{ij}, \tag{23}$$

$$\sum_{k\in\mathcal{K}} q_k x_{ij}^{ktt'} \leq u_{ij} y_{ij}^{tt'} \qquad \forall ((i,t),(j,t')) \in \mathcal{A}_\mathcal{T}, \tag{24}$$

$$\sum_{\omega\in\Omega_{ij}} \phi_\omega^k \delta_\omega + \sum_{(t,t'):((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}} x_{ij}^{ktt'} = \sum_{p\in\mathcal{P}(i,j)_k} v_p^k \qquad \forall (i,j) \in \mathcal{A}, k \in \mathcal{K}_{ij}, \tag{25}$$

$$\gamma_{ij}^k \leq T \sum_{\omega\in\Omega_{ij}} \delta_\omega \qquad \forall (i,j) \in \mathcal{A}, \tag{26}$$

$$\sum_{p\in P(i,j)_k} \alpha_{ij}^{kp} v_p^k \leq \gamma_{ij}^k + \sum_{(t,t'):(i,t),(j,t'))\in\mathcal{A}_\mathcal{T}} t x_{ij}^{ktt'} \qquad \forall (i,j) \in \mathcal{A}, k \in \mathcal{K}_{ij}, \tag{27}$$

$$\sum_{(t,t'):(i,t),(j,t'))\in\mathcal{A}_\mathcal{T}} t x_{ij}^{ktt'} + \gamma_{ij}^k \leq \sum_{p\in P(i,j)_k} \beta_{ij}^{kp} v_p^k \qquad \forall (i,j) \in \mathcal{A}, k \in \mathcal{K}_{ij}, \tag{28}$$

$$\sum_{(o_k,j)\in\mathcal{A}} \left(\gamma_{o_k j}^k + \sum_{(t,t'):((o_k,t),(j,t'))\in\mathcal{A}_\mathcal{T}} t x_{ij}^{ktt'} \right) \geq e_k \qquad \forall k \in \mathcal{K}, \tag{29}$$

$$\sum_{(j,d_k)\in\mathcal{A}} \left((\gamma_{jd_k}^k + \tau_{jd_k}(\sum_{p\in P(i,j)_k} v_p^k)) + \sum_{(t,t'):((j,t),(d_k,t'))\in\mathcal{A}_\mathcal{T}} t' x_{jd_k}^{ktt'} \right) \leq l_k \qquad \forall k \in \mathcal{K}, \tag{30}$$

$$\sum_{(i,j)\in\mathcal{A}} \left((\gamma_{ij}^k + \tau_{ij}(\sum_{p\in P(i,j)_k} v_p^k)) + \sum_{(t,t'):((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}} t' x_{ij}^{ktt'} \right) \leq \sum_{(j,i)\in\mathcal{A}} \left(\gamma_{ji}^k + \sum_{(t,t'):((j,t),(i,t'))\in\mathcal{A}_\mathcal{T}} t x_{ji}^{ktt'} \right) \qquad \forall j \in \mathcal{N}, k \in \mathcal{K}, \tag{31}$$

$$v_p^k \in \{0,1\} \qquad \forall k \in \mathcal{K}, p \in P(i,j)_k, \tag{32}$$

$$\gamma_{ij}^k \geq 0 \qquad \forall k \in \mathcal{K}, (i,j) \in \mathcal{A} \tag{33}$$

$$x_{ij}^{ktt'} \in \{0,1\} \qquad \forall ((i,t),(j,t')) \in \mathcal{A}_\mathcal{T}, k \in \mathcal{K}, \tag{34}$$

$$\delta_\omega \in \{0,1\} \qquad \forall \omega \in \Omega_{ij}, \tag{35}$$

$$z_{ij} \in \mathbb{N} \qquad \forall (i,j) \in \mathcal{A}, \tag{36}$$

$$y_{ij}^{tt'} \in \mathbb{N} \qquad \forall ((i,t),(j,t')) \in \mathcal{A}_\mathcal{T}. \tag{37}$$

The objective of the model seeks to minimize the total vehicle transportation costs, recognizing that these costs are potentially measured in two ways on each arc. Constraints (21) - (24) are the same as in the previous formulations presented. Constraints (25) ensure that if a path is chosen that contains an arc, either a consolidation is chosen for that arc or a timed version of that arc is chosen. Constraints (26) ensure that the dispatch variable for a commodity on an arc only takes a positive value if a consolidation for that arc is chosen. Constraints (29) - (31) are analogous to constraints (13) - (15), but recognize that the time at which a shipment may dispatch on a particular arc may either be represented by a continuous dispatch variable or a variable associated with a timed copy of that arc. Constraints (32) - (37) define the decision variables of the formulation and their domains.

5. Modeling additional considerations

The *TEN*($\mathcal{D}$) captures few of the operational realities considered by consolidation carriers when developing shipment and vehicle routing plans. Relatedly, the academic literature includes multiple models that capture some of these considerations. In this section we first present a variant of *CONS*($\mathcal{D}$) that captures one of the most widely-modeled considerations in the literature, empty repositioning. We next present a variant of *CONS*($\mathcal{D}$) that captures a consideration that has not yet been modeled in the literature, bin packing. The third variant integrates piecewise linear costs, which are often used in less-than-truckload transportation networks.

5.1. Asset management

The formulations presented so far presume vehicles are always available when needed. In practice, the need to move vehicles empty to position them to transport shipments can contribute greatly to the costs incurred by a carrier. The academic literature related to the SSNDP and its variants has labeled the need to move vehicles empty as a *resource management* issue. This issue can be captured in *CONS*($\mathcal{D}$), in an aggregate sense, with the following constraint.

$$\sum_{j:(i,j)\in A} z_{ij} - \sum_{j:(j,i)\in A} z_{ji} = 0 \quad \forall i \in \mathcal{N}. \tag{38}$$

This constraint ensures the number of vehicles that depart a terminal is the same as the number that arrive at that same terminal. However, as it does not consider the time at which vehicles depart and arrive a terminal it is only an approximation of empty repositioning needs. This approximation has been used in other papers (Powell and Sheffi 1989, Braklow et al. 1992) that propose formulating a service network design model on a physical network.

5.2. Bin packing

In situations wherein shipments can not be broken down and transported in different vehicles, the *TEN*($\mathcal{D}$) formulation can under-estimate capacity needs as it treats them in aggregate. To illustrate, consider a consolidation $\bar{\omega}$ on arc (i,j) consisting of three shipments k, k', k'' such that each has a size that occupies two thirds the capacity of a vehicle (e.g. $q_k = q_{k'} = q_{k''} = 2, u_{ij} = 3$). A solution to the instance of *TEN*($\mathcal{D}$) could have $x_{ij}^{ktt'} = x_{ij}^{k'tt'} = x_{ij}^{k''tt'} = 1$ for some (t,t') and thus the left-hand side of constraint (4) would have value 6, requiring $y_{ij}^{tt'} \geq 2$. Given that $y_{ij}^{tt'}$ appears with a positive coefficient in the objective function of *TEN*($\mathcal{D}$), in the absence of other inequalities an optimal solution involving this consolidation will have $y_{ij}^{tt'} = 2$. Yet no two of k, k' and k'' can fit within a single vehicle and thus three vehicles are actually needed.

Modeling this in *CONS*($\mathcal{D}$) simply requires setting $s_{\bar{\omega}} = 3$. More generally, for consolidation $\omega \in \Omega_{ij}$ we can compute the values s_ω by solving a homogeneous bin packing problem formulated with items that correspond to shipments in ω and a bin size that is u_{ij}.

However, in a setting where bin packing is to be enforced we can remove consolidations much like with the *Consolidation Pruning* procedure. Specifically, consider a consolidation ω such that $s_\omega = m > 1$. This implies that ω can be partitioned into consolidations $\omega_j, j = 1, \ldots, m$ such that $s_{\omega_j} = 1$. Because each consolidation ω_j is also used to formulate *CONS*($\mathcal{D}$), the consolidation ω can be omitted. To summarize, *CONS*($\mathcal{D}$) needs only be instantiated with consolidations ω' such that $s_{\omega'} = 1$.

5.3. Piecewise linear costs

Piecewise linear cost functions are often used in less-then-truckload transportation to model economies of scale or outsourcing rates (see eg. Balakrishnan and Graves (1989), Croxton, Gendron, and Magnanti (2003), Lapierre, Ruiz, and Soriano (2004)). Following the presentation of Frangioni and Gendron (2021), we suppose a transportation cost $g_{ij}(x_{ij})$ on an arc $(i, j) \in \mathcal{A}$ that depends on the transported quantity x_{ij} on this arc. Frangioni and Gendron (2021) presume the cost function g_{ij} is lower semi-continuous, non-decreasing and such that $g_{ij}(0) = 0$. Let us define for all $(i, j) \in \mathcal{A}$ and all $\omega \in \omega_{ij}$, the cost $\kappa^\omega_{ij} = g_{ij}(\sum_{k\in\omega} q_k)$ of dispatching all shipments in ω together on arc (i, j). Then a piecewise linear cost function can be integrated in the *CONS*($\mathcal{D}$) formulation by replacing the objective function by:

$$z_{PL\text{-}CONS(\mathcal{D})} = \sum_{(i,j)\in\mathcal{A}} \kappa^\omega_{ij}\delta_\omega.$$

Since κ^ω_{ij} can be calculated before solving the model, the objective function remains linear without extra linearization effort. In addition, no presumptions regarding the structure of the cost function $g(\cdot)$ are necessary for the correctness of this approach.

6. Computational analysis

In this section, we focus on studying the efficiency of an optimization solver when solving different formulations presented in this paper. We first study whether and when an optimization solver can solve instances of the *CONS*($\mathcal{D}$) more effectively than the *TEN*($\mathcal{D}$). We then study the performance of an optimization solver when solving instances of the hybrid consolidation-based formulation. Finally, we return to the *CONS*($\mathcal{D}$) formulation and study the performance of a solver when solving instances of models that capture empty repositioning and/or bin packing considerations.

6.1. Computational setting

To perform our computational analysis we consider a set of randomly generated instances based on a portion of the network of a United States-based LTL carrier. Specifically, a portion of the network that consists of 25 terminals (e.g. $|\mathcal{N}| = 25$) and 530 physical moves between terminals (e.g. $|\mathcal{A}| = 530$). The carrier provided cost, capacity, and travel time data.

We randomly generated instances that varied in two parameters. The first is the number of shipments, or, the size of the set $\mathcal{K}$. The second is the number of potential paths for each shipment (i.e. $|\mathcal{P}_k|$). We note that we generated instances such that all commodities have the same number of paths (i.e. $|\mathcal{P}_k| = |\mathcal{P}_{k'}|, \forall k, k' \in \mathcal{K}$). For each shipment k, to construct the set $\mathcal{P}_k$ for a given number of paths μ, we enumerated all paths from o_k to d_k to identify a set of candidate paths. We removed paths from this candidate set based on two other factors inspired by discussions with practitioners. First, we removed paths that visit more than three terminals other than the origin or destination. Second, we removed paths with distance greater than 150% the distance between origin and destination terminals, when such a direct move is

possible. From the remaining set of candidate paths, we included in $\mathcal{P}_k$ the μ shortest with respect to travel time. We refer to the number of paths per shipment in an instance by $|\mathcal{P}|$.

We define the class of an instance by the pair $(|\mathcal{K}|, |\mathcal{P}|)$. We considered classes wherein $\mathcal{K}$ takes on one of the values 100,150,200,250,or 300. Regarding paths, we consider classes wherein $|\mathcal{P}|$ takes on one of the values 5,6,7,8,9, or 10. Thus, we considered 30 classes in total. For each class we generated five instances randomly, yielding a test set that consists of 150 instances.

All experiments were run on a computer equipped with 64 Intel Xeon Gold 6130 CPU processors operating at 2.10GHz and running the Ubuntu distribution of the Linux operating system. All optimization models were solved with CPLEX 12.10 (Studio-CPLEX 2013). All formulations and code to instantiate them were implemented in Python 3.7 (VanRossum and Drake 2010). All code was run for two hours, or, 7,200 seconds. All optimization models were solved to a 1% tolerance. CPLEX 12.10 was configured to use a single thread. All other parameters of CPLEX 12.10 were left at their default values. All times reported are in seconds.

6.2. Computational comparison of TEN($\mathcal{D}$) and CONS($\mathcal{D}$)

This section focuses on the computational efficiency of solving the new formulation together with its enhancements and the extension to a hybrid formulation.

6.2.1. Comparison of basic formulations

We first study the ability of the solver to solve instances of each formulation. We present in Tables 1 and 2 the percentage of instances of each formulation in each class that could be solved within the time limit of two hours.

	$\lvert\mathcal{P}\rvert$						
$\lvert\mathcal{K}\rvert$	5	6	7	8	9	10	Average
100	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
150	100.00%	100.00%	100.00%	100.00%	80.00%	40.00%	86.67%
200	100.00%	100.00%	100.00%	40.00%	0.00%	0.00%	56.67%
250	80.00%	40.00%	40.00%	20.00%	0.00%	0.00%	30.00%
300	60.00%	0.00%	0.00%	0.00%	0.00%	0.00%	10.00%
Average	88.00%	68.00%	68.00%	52.00%	36.00%	28.00%	56.67%

Table 1: % of instances of *CONS*($\mathcal{D}$) solved

	$\lvert\mathcal{P}\rvert$						
$\lvert\mathcal{K}\rvert$	5	6	7	8	9	10	Average
100	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
150	100.00%	100.00%	100.00%	40.00%	40.00%	20.00%	66.67%
200	100.00%	100.00%	100.00%	0.00%	0.00%	0.00%	50.00%
250	20.00%	20.00%	0.00%	0.00%	0.00%	0.00%	6.67%
300	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Average	64.00%	64.00%	60.00%	28.00%	28.00%	24.00%	44.67%

Table 2: % of instances of *TEN*($\mathcal{D}$) solved

We see that more instances of *CONS*($\mathcal{D}$) were solved overall and of each class. However, we also see that for a given number of paths per commodity, as the number of commodities in an instance increases, fewer instances of *CONS*($\mathcal{D}$) are solved. Similarly, for a given $|\mathcal{K}| > 100$, as the number of paths per commodity increases, fewer instances of *CONS*($\mathcal{D}$) are solved.

To understand this further, we report in Table 3 the average number of consolidations used to formulate an instance of *CONS*($\mathcal{D}$) by instance class. Not surprisingly, we see that as either

$|\mathcal{K}|$ or $|\mathcal{P}|$ increases, the number of consolidations used to formulate the instance increases, suggesting that large numbers of consolidations yield instances of $CONS(\mathcal{D})$ that are more difficult to solve.

	$\|\mathcal{P}\|$						
$\|\mathcal{K}\|$	5	6	7	8	9	10	Average
100	1,373.20	1,751.20	2,127.00	2,444.00	2,777.20	3,166.60	2,273.20
150	3,012.20	4,196.20	6,109.00	7,398.20	8,935.20	11,711.60	6,893.73
200	6,627.00	13,716.20	32,922.20	78,534.40	96,218.60	164,180.80	65,366.53
250	12,383.40	26,273.20	55,972.40	82,530.00	113,069.40	147,035.40	72,877.30
300	25,851.20	44,499.20	103,772.60	141,768.60	191,203.20	262,254.40	128,224.87
Average	9,849.40	18,087.20	40,180.64	62,535.04	82,440.72	117,669.76	55,127.13

Table 3: Average number of consolidations per instance of $CONS(\mathcal{D})$

We next consider instances wherein both the $CONS(\mathcal{D})$ and $TEN(\mathcal{D})$ formulation could be solved within the time limit of two hours. We note that among those, instances of $CONS(\mathcal{D})$ were solved, on average, in 257.56 seconds while instances of $TEN(\mathcal{D})$ were solved in 996.54 seconds.

6.2.2. Experimental comparison of linear relaxations

To understand why the $CONS(\mathcal{D})$ is easier to solve, we solved the linear programming relaxation of the $CONS(\mathcal{D})$ and $TEN(\mathcal{D})$ formulations for each instance. We denote the objective function values of the optimal solutions to these linear programming relaxations by $z^{LPR}_{CONS(\mathcal{D})}$ and $z^{LPR}_{TEN(\mathcal{D})}$. Given these, we compute the gap $(z^{LPR}_{CONS(\mathcal{D})} - z^{LPR}_{TEN(\mathcal{D})})/z^{LPR}_{CONS(\mathcal{D})}$ as well as the average of this gap over all 180 instances. That average gap is 22.60%, suggesting that $CONS(\mathcal{D})$ is a significantly stronger formulation than $TEN(\mathcal{D})$.

We next study the strength of the two formulations from another perspective by comparing the objective function values of the respective formulations with the objective function value of the highest-quality known solution to the integer program. More specifically, letting z^{IP} represent the objective function value of that high-quality (not necessarily optimal) solution, we compute for $CONS(\mathcal{D})$ the gap $IPGap_{CONS(\mathcal{D})} = (z^{IP} - z^{LPR}_{CONS(\mathcal{D})})/z^{IP}$. The gap $IPGap_{TEN(\mathcal{D})}$ is computed similarly. In Table 4 we report the averages of these gaps over all instances with the same number of shipments. We see that not only is $CONS(\mathcal{D})$ a much stronger formulation but that its linear programming relaxation provides a bound that is often within 5% of the best known solution to the integer program.

	$\|\mathcal{K}\|$					
	100	150	200	250	300	Average
$IPGap_{CONS(\mathcal{D})}$	2.36%	4.33%	3.37%	4.11%	5.14%	3.86%
$IPGap_{TEN(\mathcal{D})}$	22.37%	27.82%	26.44%	26.25%	23.78%	25.33%

Table 4: LP gaps with best-known IP solutions

6.2.3. Consolidation Pruning

We next consider the proposed speed-up techniques. We report in Table 5 the percentage of instances of $CONS(\mathcal{D})$ that could be solved within the time limit of two hours when consolidation pruning is used and when it is not. Then, of the instances that could be solved in both cases, we report the average time needed in each case. Finally, of the instances that could not be solved either with or without consolidation pruning, we report the average optimality gap in each case. We see that the use of *Consolidation Pruning* enables more instances to be solved and in less time. Relatedly, of those not solved, the average optimality gap is smaller.

	% instances solved	Time	Opt. gap not solved
With *CP*	60.67%	400.69	3.84%
Without	56.67%	692.75	5.33%

Table 5: Solving *CONS*($\mathcal{D}$) with and without *Consolidation Pruning*

As another measure of the effectiveness of *Consolidation Pruning* we present in Table 6 the average number of consolidations used to formulate an instance of *CONS*($\mathcal{D}$), both when *Consolidation Pruning* is used and when it is not used. We see that particularly for instances with larger numbers of commodities the pruning procedure removes the vast majority of consolidations.

	$\lvert\mathcal{K}\rvert$					
	100	150	200	250	300	Average
With *CP*	1,660.90	3,890.97	8,218.43	13,959.37	21,600.13	9,952.67
Without	2,273.20	6,893.73	65,366.53	72,877.30	128,224.87	55,127.13
# With/# Without	73.06%	24.09%	2.54%	2.28%	1.30%	18.05%

Table 6: Average number of consolidations per instance of *CONS*($\mathcal{D}$)

6.2.4. Valid inequalities

We next turn our attention to the valid inequalities discussed in Section 3.2, using *CONS*($\mathcal{D}$) with consolidation pruning as a baseline. We consider when only the *Origin Consolidation Cutset* (OCC) and *Destination Consolidation Cutset* (DCC) inequalities are added to the formulation as well as when they and the *Vehicle Cutset* (VC) inequalities are added.

We report in Table 7 for each configuration the percentage of all instances that could be solved within the two hour time limit. Then, of the instances that could be solved for each configuration we report the average time needed to do so. Finally, of the instances that could not be solved by any configuration, we report the average optimality gap.

	% instances solved	Time	Opt. gap not solved
With *CP*	60.67%	510.85	4.03%
+ OCC and DCC inequalities	62.67%	455.89	3.43%
+ VC inequalities	59.33%	726.28	5.04%

Table 7: Effectiveness of valid inequalities

We see that adding the OCC and DCC inequalities enables the solver to solve more instances while adding the VC inequalities as well does not. We hypothesize that a reason adding the VC inequalities does not increase the number of instances solved is that the constraints in the *CONS*($\mathcal{D}$) formulation that model capacity usage (e.g. the constraints $z_{ij} \geq \sum_{\omega \in \Omega_{ij}} s_\omega \delta_\omega$) are much stronger than those present in *TEN*($\mathcal{D}$) (e.g. the constraints $\sum_{k \in \mathcal{K}} q_k x_{ij}^{ktt'} \leq u_{ij} y_{ij}^{tt'}$).

6.2.5. Solving the hybrid formulation

In this section, we study the ability of the solver to solve instances of the hybrid formulation presented in Section 4. Instantiating the hybrid formulation requires partitioning the time interval for each arc $(i,j) \in \mathcal{A}$ into time intervals $(\mathcal{C}(i,j)_q)$ during which shipment dispatch times are modeled with continuous dispatch variables and time intervals $(\mathcal{E}(i,j)_q)$ during which they are modeled as with a time-expanded network. To perform this partition for arc $(i,j) \in \mathcal{A}$, we assume a parameter ν that dictates the maximum cardinality consolidation to consider in the formulation. Then, for any consolidation $\omega \in \Omega_{ij}$ such that $|\omega_{ij}| > \nu$, the corresponding time window during which that consolidation may occur is used to define the intervals during

which shipment dispatch times are modeled as with a time-expanded network (i.e. the sets $\mathcal{E}(i,j)_q$). The remaining time intervals are used to construct the sets $\mathcal{C}(i,j)_q$.

To study the strength of the hybrid formulation, we illustrate in Figure 2 the percentage of instances that could be solved for values of ν ranging from 4 to 10. We note that the *Consolidation Pruning* procedure was not used in the experiments conducted to generate these results. Nor were the OCC or DCC inequalities added to the formulation.

We observe that for all values of ν considered in these experiments at least as many instances of the resulting hybrid formulation were solved as instances of the *CONS*($\mathcal{D}$). However, for $\nu = 5$, nearly 9% more instances of the hybrid formulation were solved than *CONS*($\mathcal{D}$).

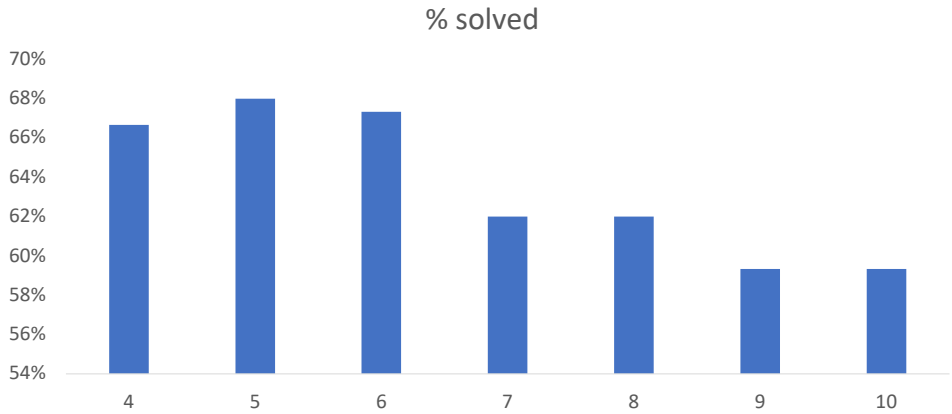

Figure 2: Percentage of instances of hybrid formulation solved, by ν, the maximum cardinality consolidation

Continuing the comparison between *CONS*($\mathcal{D}$) and the hybrid formulation we present in Table 8 the percentage of instances of each formulation solved, the average time to solve instances wherein each formulation could be solved, and the average gap of those instances wherein neither formulation could be solved. We see that in addition to enabling more instances to be solved, for the instances for which both formulations could be solved, the hybrid formulation instances were solved in much less time. Relatedly, for the instances for which neither formulation could be solved, the average optimality gap reported at termination of solving the hybrid formulation is much smaller. Finally, we note that over all the instances of the hybrid formulation with

	% instances solved	Time	Opt. gap not solved
CONS($\mathcal{D}$)	59.33%	692.75	5.33%
Hybrid, $\nu = 5$	68.00%	164.30	2.20%

Table 8: Comparision of *CONS*($\mathcal{D}$) and hybrid formulation

$\nu = 5$ that could be solved, the average time to do so was 715.10 seconds. Relatedly, of all the instances of the hybrid formulation with $\nu = 5$ that could not be solved, the average optimality gap reported at termination of the solver was 2.15%.

To further understand why solving the hybrid formulation is effective we report in Table 9 the average gaps between the optimal objective function value of the linear programming relaxation of *TEN*($\mathcal{D}$) and the *CONS*($\mathcal{D}$) as well as the hybrid formulation. Relatedly, we report in Table 10 the percentage of arcs that are modeled as hybrid arcs. In other words, arcs wherein $\mathcal{E}(i,j)_1 \neq \emptyset$.

Comparing the two tables we see that the hybrid formulation is significantly stronger than the *TEN*($\mathcal{D}$) but, not surprisingly, weaker than the *CONS*($\mathcal{D}$). Also not surprisingly, the strength of the hybrid formulation as compared to *CONS*($\mathcal{D}$) depends on the percentage of

	$\lvert\mathcal{K}\rvert$					
	100	150	200	250	300	Average
CONS($\mathcal{D}$)	20.50%	24.57%	23.87%	23.08%	20.33%	22.47%
Hybrid, $\nu = 5$	20.07%	23.21%	21.46%	19.35%	15.76%	19.97%

Table 9: Gap with linear programming relaxation of *TEN*($\mathcal{D}$)

	$\lvert\mathcal{K}\rvert$					
	100	150	200	250	300	Average
% hybrid arcs, $\nu = 5$	0.77%	3.13%	5.94%	9.72%	13.50%	6.61%

Table 10: Percentage of hybrid arcs in $\mathcal{D}$, by $\lvert\mathcal{K}\rvert$

the arcs in the network that are modeled in a hybrid fashion, and that percentage increases as the number of shipments associated with an instance increases.

We next consider solving the hybrid formulation with *Consolidation Pruning* and the OCC and DCC valid inequalities. We report solving statistics in Table 11 and see that with these additional techniques, nearly 73% of instances are solved, and in less than 15 minutes of time (on average). In addition, the optimality gap associated with instances that were not solved is small.

	$\lvert\mathcal{K}\rvert$					
	100	150	200	250	300	Average
% solved	100.00%	100.00%	90.00%	53.33%	20.00%	72.67%
Solve time	3.92	349.83	1,136.10	2,329.46	1,397.56	797.65
Gap not solved			1.06%	1.90%	3.08%	2.52%

Table 11: Solving hybrid formulation, $\nu = 5$, with CP and OCC, DCC valid inequalities.

Finally, we recall that a motivation for the hybrid formulation was to avoid creating instances with large numbers of consolidations. To assess whether this was achieved, we report in Table 12 the average number of consolidations created in instances of four different formulations, averaged over instances with the same number of shipments. We repeat statistics for *CONS*($\mathcal{D}$) and *CONS*($\mathcal{D}$)+CP in Table 12 for ease of comparison. We see that instantiating the hybrid formulation requires far fewer consolidations than *CONS*($\mathcal{D}$) both when consolidation pruning is applied and when it is not.

6.2.6. Synthesis

Finally, we summarize the computational efficiency of solving different formulations in Table 13. In this table we report, by number of shipments in an instance, the percentage of instances solved to within a 1% optimality tolerance. We also report for each formulation the average time it took to solve the instances of that formulation that could be solved. Note that in Table 13, and unlike previous tables, the set of instances being averaged over varies by formulation. For example, the average for *TEN*($\mathcal{D}$) is over a different (smaller) set of instances than the average for *CONS*($\mathcal{D}$). Finally, we report for each formulation the average optimality gap of instances of that formulation that could not be solved. Similarly, this average is computed over a set of instances that varies by formulation. The average for *TEN*($\mathcal{D}$) is over a larger set of instances than the average for *CONS*($\mathcal{D}$).

We see that by solving instances of the *CONS*($\mathcal{D}$) formulation, along with the proposed speed-up techniques, 17% more can be solved than by solving the *TEN*($\mathcal{D}$) formulation. An additional 11% more instances of the hybrid formulation can be solved when the proposed

	$\lvert\mathcal{K}\rvert$					
	100	150	200	250	300	Average
CONS($\mathcal{D}$)	2,273.20	6,893.73	65,366.53	72,877.30	128,224.87	55,127.13
Hybrid($\nu = 5$)	2,182.37	5,560.48	10,588.10	20,209.60	32,623.26	14,232.76
CONS($\mathcal{D}$)+CP	1,660.90	3,890.97	8,218.43	13,959.37	21,600.13	9,952.67
Hybrid($\nu = 5$) +CP+OCC&DCC	1,588.83	3,162.77	4,478.00	6,739.27	8,361.17	4,866.01

Table 12: Average number of consolidations.

	$\lvert\mathcal{K}\rvert$						Time	Opt. gap
	100	150	200	250	300	Average	solved	not solved
TEN($\mathcal{D}$)	100.00%	66.67%	50.00%	6.67%	0.00%	44.67%	996.54	4.04%
CONS($\mathcal{D}$)	100.00%	86.67%	56.67%	30.00%	10.00%	56.67%	692.75	5.33%
CONS($\mathcal{D}$)+CP	100.00%	96.67%	60.00%	36.67%	10.00%	60.67%	761.19	3.79%
CONS($\mathcal{D}$) +CP + OCC&DCC	100.00%	96.67%	63.33%	36.67%	11.67%	61.67%	784.84	3.58%
Hybrid($\nu = 5$)	100.00%	93.33%	86.67%	40.00%	20.00%	68.00%	715.10	2.15%
Hybrid($\nu = 5$) +CP+OCC&DCC	100.00%	100.00%	90.00%	53.33%	20.00%	72.67%	797.65	2.52%

Table 13: Synthesis of experiments

speed-up techniques are used. While the hybrid formulation is easier to solve, as we will see in the next section, the pure *CONS*($\mathcal{D}$) formulation allows for more accurate modeling of complex operational issues such as bin packing.

6.3. Empty repositioning and bin-packing

In this section we first consider the ability to solve instances of models that explicitly capture aggregate empty repositioning constraints. We then consider instances of models wherein bin packing considerations are explicitly considered.

Regarding empty repositioning, we consider instances of the hybrid formulation wherein the aggregate empty repositioning constraints (38) are also included. We instantiate these hybrid formulations based on the parameter value $\nu = 5$ and with the OCC and DCC inequalities. We also apply *Consolidation Pruning* to the instances under consideration. We consider the same instances reported on in the previous section. We report in Table 14 three statistics, averaged over instances with the same number of shipments. We report for each number of shipments the percentage of instances solved, the average time needed to do so, and the optimality gap of instances that were not solved. Comparing Table 14 with Table 11 we see that adding the empty repositioning constraints (38) to the *CONS*($\mathcal{D}$) formulation leads to instances that are slightly harder to solve.

	$\lvert\mathcal{K}\rvert$					
	100	150	200	250	300	Average
% solved	100.00%	100.00%	86.67%	46.67%	23.33%	71.33%
Solve time	3.97	255.74	907.92	1,850.45	2,174.18	677.78
Gap not solved		1.22%	1.82%	2.62%	2.19%	

Table 14: Solving hybrid formulation instances that recognize empty repositioning

We next turn to instances of *CONS*($\mathcal{D}$) that capture bin packing considerations, as discussed in Section 5.2. We note that in this analysis we consider additional classes of instances. Namely, those with 350,400, 450, and 500 shipments. Instances in each of these classes are generated in the same manner as those already discussed. We report in Table 15 the same statistics reported in Table 14. Comparing Table 15 with Tables 1 and 5, we see that larger instances of such models can be solved and in less time, on average. We also see that of the instances that could not be solved, the optimality gap is typically smaller than when bin packing is not considered.

	$\lvert\mathcal{K}\rvert$									
	100	150	200	250	300	350	400	450	500	Average
	100	150	200	250	300	350	400	450	500	Average
% solved	100.00%	100.00%	100.00%	86.67%	90.00%	86.67%	76.67%	60.00%	26.67%	80.74%
Solve time	1.32	34.07	69.09	967.39	914.52	2,348.06	2,453.13	2,887.43	1,070.23	1,059.57
Gap not solved				1.67%	1.16%	1.59%	1.89%	2.00%	2.73%	2.19%

Table 15: Solving instances of bin packing models

Finally, we recall that ignoring bin packing can lead to underestimates of capacity needs and transportation costs. To understand the magnitude of this underestimate, we consider the gap between objective function values of high-quality solutions to $CONS(\mathcal{D})$ formulated with and without bin packing considerations. More precisely, we let z_{NO-BP} represent the objective function value of a high-quality solution to $CONS(\mathcal{D})$ formulated without bin packing. Similarly, we let z_{BP} represent the objective function value of a high-quality solution to $CONS(\mathcal{D})$ formulated with bin packing. We then compute $gap = (z_{BP} - z_{NO-BP})/z_{NO-BP}$. We report this gap, averaged over instances with the same number of shipments, in Table 16. We see that this gap is significant, over 5%. In addition, this gap increases as the number of shipments associated with an instance increases.

	$\lvert\mathcal{K}\rvert$					
	100	150	200	250	300	Average
% increase in cost	2.60%	4.36%	6.16%	7.21%	7.48%	5.56%

Table 16: Percentage increase in costs when recognizing bin packing, by $\lvert\mathcal{K}\rvert$

7. Conclusion

In this paper, we proposed new formulations for the SSNDP based on formulating with consolidations of shipments. Doing so obviates the need to use a time-expanded network to capture the synchronization of shipments needed for their consolidation. Instead, one can formulate on the physical network itself, which leads to much smaller and less symmetric integer programs. Formulating with consolidations also enables explicit representation of vehicle capacity needs, resulting in much stronger linear programming relaxations. Computationally, instances of the proposed consolidation-based formulations are easier to solve than those based on a time-expanded network. Relatedly, the consolidation-based formulation enables instances to be solved that previously would have required implementing powerful but elaborate algorithm such as *Dynamic Discretization Discovery*. Formulating with consolidations also facilitates modeling issues that have not yet been addressed in the literature, such as bin-packing considerations when computing vehicle capacity needs. In addition, the proposed modeling technique for bin-packing considerations in a consolidation based formulation yields instances that are easier to solve than those wherein capacity is modeled in an aggregate sense.

The proposed formulations present multiple avenues for future research. From an algorithmic perspective, the enumerative nature of modeling with consolidations points towards the need for a solution method based on branch-and-price. However, the efficiency we observed when solving instances of the hybrid formulation suggests a *Dynamic Discretization Discovery*-type approach for discretized time windows may be promising for large scale problems.

From a modeling perspective, formulating with consolidations makes capturing other phenomenon that are prevalent in logistics much easier, in terms of solving the resulting integer program, than with a time-expanded network-based formulation. For instance, it is common in freight transportation that some products cannot travel together in a vehicle. Modeling this

with a consolidation-based formulation is straightforward. Similarly, vehicles may have multiple types of compartments, each with a different storage capability (e.g. different temperatures), and capacities quoted along multiple dimensions. This can be integrated into a consolidation-based formulation in a manner similar to bin packing, albeit for each compartment.

References

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

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

Bakir I, Erera A, Savelsbergh M, 2021 *Motor Carrier Service Network Design.* Crainic TG, Gendreau M, Gendron B, eds., *Network Design with Applications to Transportation and Logistics*, 427–467 (Springer).

Balakrishnan A, Graves SC, 1989 *A composite algorithm for a concave-cost network flow problem. Networks* 19(2):175–202.

Barnhart C, Schneur R, 1996 *Network design for express freight service. Operations Research* 44:852–863.

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

Boland N, Hewitt M, Marshall L, Savelsbergh M, 2019 *The price of discretizing time: a study in service network design. EURO Journal on Transportation and Logistics* 8(2):195–216.

Braklow JW, Graham WW, Hassler SM, Peck KE, Powell WB, 1992 *Interactive optimization improves service and performance for Yellow Freight System. Interfaces* 22(1):147–172.

Chouman M, Crainic TG, 2015 *Cutting-plane matheuristic for service network design with design-balanced requirements. Transportation Science* 49(1):99–113.

Chouman M, Crainic TG, 2021 *Freight Railroad Service Network Design.* Crainic TG, Gendreau M, Gendron B, eds., *Network Design with Applications to Transportation and Logistics*, 381–424 (Springer).

Chouman M, Crainic TG, Gendron B, 2017 *Commodity representations and cut-set-based inequalities for multicommodity capacitated fixed-charge network design. Transportation Science* 51(2):650–667.

Christiansen M, Hellsten E, Pisinger D, Sacramento D, Vilhelmsen C, 2021 *Liner Shipping Service Network Design.* Crainic TG, Gendreau M, Gendron B, eds., *Network Design with Applications to Transportation and Logistics*, 467–504 (Springer).

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

Crainic TG, Gendreau M, 2021 *Heuristics and metaheuristics for fixed-charge network design.* Crainic TG, Gendreau M, Gendron B, eds., *Network Design with Applications to Transportation and Logistics*, 89–138 (Springer).

Crainic TG, Gendron B, 2021 *Exact methods for fixed-charge network design.* Crainic TG, Gendreau M, Gendron B, eds., *Network Design with Applications to Transportation and Logistics*, 29–89 (Springer).

Crainic TG, Hewitt M, 2021 *Service Network Design.* Crainic TG, Gendreau M, Gendron B, eds., *Network Design with Applications to Transportation and Logistics*, 347–382 (Springer).

Crainic TG, Hewitt M, Toulouse M, Vu DM, 2016 *Service network design with resource constraints. Transportation Science* 50(4):1380–1393.

Croxton KL, Gendron B, Magnanti TL, 2003 *A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems. Management Science* 49(9):1268–1273.

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

Frangioni A, Gendron B, 2021 *Piecewise linear cost network design.* Crainic TG, Gendreau M, Gendron B, eds., *Network Design with Applications to Transportation and Logistics*, 167–185 (Springer).

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

Hewitt M, Crainic TG, Nowak M, Rei W, 2019 *Scheduled service network design with resource acquisition and management under uncertainty. Transportation Research Part B: Methodological* 128:324–343.

Hewitt M, Lehuédé F, 2022 *The service network scheduling problem.* Technical report, URL `https://hal.archives-ouvertes.fr/hal-03598983/`.

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

Lapierre SD, Ruiz AB, Soriano P, 2004 *Designing distribution networks: Formulations and solution heuristic. Transportation Science* 38(2):174–187.

Margot F, 2010 *Symmetry in integer linear programming. 50 Years of Integer Programming 1958-2008*, 647–686 (Springer).

Marshall L, Boland N, Savelsbergh M, Hewitt M, 2020 *Interval-based dynamic discretization discovery for solving the continuous-time service network design problem. Transportation Science* .

Nemhauser GL, Wolsey LA, 1988 *Integer and combinatorial optimization*, volume 18 (Wiley New York).

Pfetsch ME, Rehn T, 2019 *A computational comparison of symmetry handling methods for mixed integer programs. Mathematical Programming Computation* 11(1):37–93.

Powell WB, Sheffi Y, 1989 *Design and Implementation of an Interactive Optimization System for Network Design in the Motor Carrier Industry. Operations Research* 37:12–29.

Scherr YO, Saavedra BAN, Hewitt M, Mattfeld DC, 2019 *Service network design with mixed autonomous fleets. Transportation Research Part E: Logistics and Transportation Review* 124:40–55.

Schulz JD, 2021 *Rival carriers chase former central freight customers as tight ltl market shrinks.* URL `https://www.logisticsmgmt.com/article/rival_carriers_chase_former_central_freight_customers_as_tig`

Studio-CPLEX IICO, 2013 *Users manual-version 12 release 6.*

Teypaz N, Schrenk S, Cung VD, 2010 *A decomposition scheme for large-scale service network design with asset management. Transportation Research Part E: Logistics and Transportation Review* 46(1):156–170.

UPS, 2021 *Ups annual report.* URL `https://investors.ups.com/company-profile`.

VanRossum G, Drake FL, 2010 *The python language reference* (Python Software Foundation Amsterdam, Netherlands).

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

Zhu E, Crainic TG, Gendreau M, 2014 *Scheduled service network design for freight rail transportation. Operations research* 62(2):383–400.