



The Service Network Scheduling Problem

Mike Hewitt, Fabien Lehuédé

► To cite this version:

| Mike Hewitt, Fabien Lehuédé. The Service Network Scheduling Problem. 2022. hal-03710468v4

HAL Id: hal-03710468

<https://hal.science/hal-03710468v4>

Preprint submitted on 4 Oct 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.

The Service Network Scheduling Problem

Mike Hewitt¹

Fabien Lehu  d  ²

¹Department of Information Systems and Supply Chain Management,
Quinlan School of Business, Loyola University Chicago,
`mhewitt3@luc.edu`

²IMT Atlantique, LS2N, UMR CNRS, Nantes, France,
`fabien.lehuede@imt-atlantique.fr`

Abstract

We consider the optimization problem of determining schedules for shipments on known paths within a terminal network in order to minimize vehicle transportation costs. We refer to this problem as the *Service Network Scheduling Problem* and present two mixed integer programming formulations of that problem. The first is based on the classical idea of a time-expanded network. The second formulation is new and is based on sets of shipment consolidations. We show both analytically and computationally that the consolidation-based formulation can be the superior of the two, but that its enumerative nature renders it ineffective for instances with large numbers of shipments. Thus, we also present a column generation-based algorithm for solving the consolidation-based formulation that relies on solving relaxations that are integer programs. We demonstrate the superior performance of this algorithm with a computational study wherein we compare it against applications of state-of-the-art approaches from the literature.

1 Introduction

Consolidation carriers are transportation companies that transport shipments that are small relative to vehicle capacity. Consolidation carrier is an umbrella term that covers companies participating in one (or both) of the Less-than-truckload (LTL) freight and small package/parcel industries. Both industries are large; 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). Many of these companies execute very large scale operations. For example, UPS reports delivering 6.3 billion packages in 2020 (UPS 2021).

As transportation exhibits economies of scale, profitability for such carriers is driven by consolidation. Specifically, dispatching vehicles that transport multiple shipments, each potentially associated with a different customer. Such consolidation is typically enabled by routing shipments on paths through a network of terminals as opposed to directly from customer origin to destination, as is often done in full truckload transportation. For large-scale operations, small percentage improvements in consolidation (typically measured by vehicle utilization) can lead to significant reductions in transportation costs.

Among the decisions that are taken by carriers to plan their network, a *path* has to be determined for each shipment. A path is defined by the sequence of terminals it visits, with the sequence beginning at the origin terminal for the shipment and ending at its destination terminal. Then, a *schedule* has to be determined for the path of each shipment. This schedule determines the time at which each leg of the path is executed. Finally, *transportation moves* of capacitated vehicles are planned to operate these transportation legs. Transportation moves executed by a vehicle incur a cost that is independent of the shipments it transports. Achieving high levels of consolidation of shipments in vehicles, and low transportation costs, requires determining paths for shipments through such networks, and schedules for those paths, that enable multiple shipments to dispatch on the same transportation move at the same time. The determination of such paths and schedules can often be assisted by solving some variant of the *Scheduled Service Network Design Problem* (SSNDP) (Crainic 2000, Jarrah et al. 2009, Zhu et al. 2014, Boland et al. 2017a,b, Hewitt 2019, Hewitt et al. 2019, Marshall et al. 2020).

We consider the case of a LTL freight transportation carrier that is reluctant to frequently changing shipment paths. Generally speaking, consistency in operations reduces complexity and errors and increases service quality. More specifically, the determination of shipment paths in a network can then trigger additional planning processes such as the assignment of shipment destination terminals to outbound

doors in a cross-dock. Changing paths may necessitate changing these assignments. Some terminals may be equipped with information systems that facilitate changing these assignments with few subsequent operational errors. Others may not, in which case frequent changes may lead to shipments being loaded into vehicles at the wrong outbound door. Maintaining shipment paths, and hence door assignments, does leave flexibility with respect to the dispatch times of vehicles (scheduling of paths), especially if it allows to reduce the number of vehicle moves and transportation costs.

Hence, after solving a SSNDP based on a forecast, the carrier seeks to keep shipment paths constant over some planning horizon (e.g. a week), in order to maintain consistency with the operations that support the execution of those paths. However, in the near-term when more accurate estimates of shipment volumes are known, the schedules of those paths are adjusted to maximize consolidation. Thus, we consider the problem of optimally determining shipment path schedules in order to minimize total vehicle transportation costs. We refer to this problem as the *Service Network Scheduling Problem* (SNSP).

To summarize, both the SSNDP and SNSP determine the number of vehicles to be dispatched on each transportation leg and the times of those dispatches. The SSNDP also determines a path and schedule for each commodity on the network. The SNSP keeps the paths taken by each commodity fixed and designs a new schedule for those paths. The SNSP can be seen as an operational adjustment of a load plan generated by solving a SSNDP at a tactical level. In addition to being relevant to practice, the SNSP can also appear as a subproblem of exact or heuristic algorithms for solving the SSNDP.

We propose two mixed integer programming formulations of the SNSP. The first is based on the classical idea of a time-expanded network (Ford and Fulkerson 1958, 1962). The second is based on enumerations of consolidations of shipments, wherein a consolidation of shipments represents a set of shipments that are transported on the same move at the same time. We prove the equivalence of the formulations. We also show both analytically and computationally that the consolidation-based formulation can be the stronger of the two, but that its enumerative nature renders it computationally ineffective for instances based on large numbers of shipments.

Thus, we present an algorithm for solving the consolidation-based formulation that does not enumerate sets of consolidations *a priori*, but instead dynamically in the course of its execution. Clearly, one framework for such an algorithm is Branch-and-Price (Barnhart et al. 1998, Desaulniers et al. 2006). As such an algorithm involves solving linear relaxations of a formulation, many of the technical advancements of today’s solvers at solving integer programs is lost. We present an algorithm that operates in a fashion that is similar to Branch-and-Price, in that at each iteration it solves an optimization problem and then uses information from the solution to that problem to determine variables to add to the optimization problem. It differs from Branch-and-Price in that the optimization problem solved at an iteration is an integer program that is formulated in such a manner as to be a relaxation of the original problem. We refer to this algorithm as *IP – ColGen*, or, Integer Programming-based Column Generation. The idea of Integral Column Generation has been applied to other transportation applications (Tahir et al. 2021). We prove the correctness of this algorithm and demonstrate with a computational study based on a US-based LTL carrier its ability to solve instances derived from real-world operations in much less time than applications of state-of-the-art methods from the literature.

We believe this paper presents multiple contributions to the literature on optimizing freight transportation. First, to the best of our knowledge, this is the first paper to study the SNSP. Second, we present a consolidation-based formulation of the SNSP that is new to the literature and computationally superior to an approach based on classical techniques. Third, we present an algorithm for solving this consolidation-based formulation that is very effective computationally in comparison to state-of-the-art methods.

The rest of this paper is organized as follows. Section (2) discusses literature relevant to the problem we consider. Section (3) describes the planning problem we consider in greater detail and presents two mixed integer programming formulations of that problem. Section (3.4) presents a comparison of the two formulations. Section (4) presents an algorithm for solving the consolidation-based formulation while Section (5) presents a computational analysis of the performance of the proposed algorithm with other methods. Section (6) concludes the paper and outlines future research directions extending the ideas presented in this paper.

2 Literature review

Much of the academic literature relevant to planning freight transportation operations in a terminal network focuses on determining paths and schedules by solving some variant of the *Scheduled Service Network Design Problem* (SSNDP) (Crainic 2000, Jarrah et al. 2009, Zhu et al. 2014, Boland et al.

2017a,b, Hewitt 2019, Hewitt et al. 2019, Marshall et al. 2020). A related problem, the *Service Network Design Problem* (SNDP) determines shipment paths but not schedules (Wieberneit 2008, Andersen et al. 2009, 2011, Chouman and Crainic to appear, Baubaid et al. 2020). Both the SSNDP and SNDP have primarily been used to plan long haul freight operations.

The literature typically views solving the SSNDP (and SNDP) as part of a tactical planning process, meaning shipment paths and schedules are presumed fixed over some medium-term planning horizon (Jarrah et al. 2009, Erera et al. 2013, Zhu et al. 2014). Some literature (Crainic and Sgalambro 2014, Scherr et al. 2019, 2020) considers the applicability of these problems to urban logistics. Some literature (Andersen et al. 2009, 2011, Crainic et al. 2014, Hewitt et al. 2019) expands the decision-making of the SSNDP to include asset management considerations regarding vehicles such as the need to move a vehicle empty to position it for a future move. As the SNSP focuses only on scheduling paths, it considers a subset of the decisions prescribed by the SSNDP.

If one views vehicle transportation moves as machines, the SNSP is similar in nature to a *Flow Shop Scheduling Problem* (FSSP) (Pinedo 2012). However, in the FSSP the number of machines is given, whereas our problem seeks to minimize an objective that is a function of the number of machines used. The formulation we propose for the SNSP is similar in spirit to the optimization problem solved in *Dynamic Discretization Discovery* (Boland et al. (2017a)) to verify whether a given set of commodity paths and consolidations of commodities on arcs of those paths is feasible. It is also similar in spirit to the extended formulation of the *Generalized Assignment Problem* (GAP) presented in Savelsbergh (1997). It is different in that the SNSP involves scheduling a sequence of activities (e.g. vehicle and shipment dispatches).

The algorithm we propose for solving the SNSP follows a classical framework wherein a relaxation of the problem is iteratively solved and refined until a solution to the relaxation is feasible (and hence optimal) to the original problem. Examples of applications of this framework include cutting plane algorithms such as Benders decomposition (Benders 1962) and Branch-and-Cut (Nemhauser and Wolsey 1988). The algorithm we propose is similar to Branch-and-Price (Barnhart et al. 1998, Desaulniers et al. 2006) in that it solves an optimization problem that is formulated (in most iterations) over a subset of the variables in the original problem. The algorithm we propose is different from Branch-and-Price in that the optimization problem it solves is an integer program that is formulated to be a relaxation of the original problem. In this way, it is similar to the *Dynamic Discretization Discovery* (DDD) algorithm proposed in (Boland et al. 2017a)). The algorithm we propose is different from DDD in that the domains of the decision variables remain constant throughout its execution. It is also similar to Integral Column Generation (Rönnerberg and Larsson 2009, 2014, Tahir et al. 2019, 2021, 2022). It is different in that it is not based on simplex-type pivots.

3 Problem description and mathematical programming formulations

In this section, we first present a description of the transportation problem we seek to solve as well as modeling constructs and notation common to the two mixed integer programming formulations of that problem we propose. We then present those formulations. We finish with an analysis of the relative strength of the two formulations.

3.1 Problem description and mathematical notation

We consider a company that transports shipments from customer origin to customer destination. The transportation network of the company can be defined by a directed graph $\mathcal{D} = (\mathcal{N}, \mathcal{A})$, where \mathcal{N} is a set of nodes that model terminals in the network and \mathcal{A} is a set of arcs that model transportation moves within that network. Formally, the set \mathcal{A} consists of pairs $(i, j), i, j \in \mathcal{N}$ that model physical travel from terminal i to terminal j . We consider that the network is operated by a homogeneous fleet of vehicles with identical capacities, speed and cost. Associated with each arc $(i, j) \in \mathcal{A}$ is a travel time denoted by τ_{ij} , a per-vehicle capacity denoted by u_{ij} , and a per-vehicle cost denoted by f_{ij} .

We presume that each shipment is associated with a commodity $k \in \mathcal{K}$, with \mathcal{K} the set of all shipments/commodities that must be routed through the network. Associated with each commodity $k \in \mathcal{K}$ is an origin terminal, o_k , and destination terminal, d_k . We let e_k denote the earliest time at which commodity k is available at its origin terminal, o_k , and l_k denote the latest time at which k can be delivered

to its destination terminal, d_k . Lastly, associated with commodity k is the shipment size denoted q_k , quoted in the same unit as the vehicle capacity u_{ij} , $(i, j) \in \mathcal{A}$ associated with arcs.

In the problem we study, we presume a single path is given for each shipment. We let $p_k = v_1^k, \dots, v_{r_k}^k$ represent the sequence of nodes from \mathcal{N} in the path of commodity $k \in \mathcal{K}$, with $v_1^k = o_k$ and $v_{r_k}^k = d_k$. We let $\mathcal{P} = \cup_{k \in \mathcal{K}} \mathcal{P}_k$ denote the set of all such shipment paths.

We seek to schedule the dispatch time of each shipment on each move in its path, and schedule vehicle dispatches on moves to support shipment dispatches. We presume that a shipment must dispatch entirely at the same time on each move in its path. More specifically, the problem seeks to determine the times at which each shipment dispatches on each of the moves on its path, while ensuring that those dispatch times agree with the shipment earliest available and latest due times, and the travel times for those moves. In addition, the problem determines the dispatch times of vehicles with sufficient capacity to transport those shipments. We presume that the time horizon during which commodity paths are to be fixed can be discretized according to a time step (of e.g. 15 minutes) for which prescribed decisions can be executed. The objective is to minimize the total transportation cost associated with the vehicle moves needed to transport a known set of shipments over a fixed planning horizon. We refer to this problem as the *Service Network Scheduling Problem* (SNSP).

As complementary notation, for each commodity $k \in \mathcal{K}$ we let the node set $\mathcal{N}^k \subseteq \mathcal{N}$ contain the nodes v_i^k in that path and the arc set $\mathcal{A}^k \subseteq \mathcal{A}$ contain the arcs (v_i^k, v_{i+1}^k) in that path. We also let $\mathcal{K}_{ij} = \{k \in \mathcal{K} : (i, j) \in \mathcal{A}^k\}$ denote the set of commodities with a path that contains arc $(i, j) \in \mathcal{A}$. Lastly, we note that given the path p_k and earliest available and latest due times e_k, l_k for commodity k , one can derive a time window $[\alpha_v^k, \beta_v^k]$ during which k can be at each node v in its path.

Having defined notation that is common to our two formulations of the SNSP, we next present a mixed integer programming formulation that is based on a time-expanded network.

3.2 Time-expanded network formulation

For this formulation, we presume a discretization of time, $\mathcal{T} = \{1, \dots, T\}$, of a planning horizon that is T periods long. We model shipment and vehicle movements in both space and time with the directed network $\mathcal{D}_{\mathcal{T}} = (\mathcal{N}_{\mathcal{T}}, \mathcal{A}_{\mathcal{T}} \cup \mathcal{H}_{\mathcal{T}})$. Here, $\mathcal{N}_{\mathcal{T}}$ 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 arcs of the form $((i, t), (j, t'))$, $(i, t), (j, t') \in \mathcal{N}_{\mathcal{T}}$, $i \neq j$, $t' = \min(t'' \in \mathcal{T} | t'' - t \geq \tau_{ij})$ and model traveling from terminal i at time t to arrive at terminal j at time t' . The set $\mathcal{H}_{\mathcal{T}}$ consists of arcs of the form $((i, t), (i, t+1))$, $(i, t), (i, t+1) \in \mathcal{N}_{\mathcal{T}}$ and model waiting at terminal i from time t to time $t+1$. Like above, we let the sets $\mathcal{N}_{\mathcal{T}}^k = \{(i, t) \in \mathcal{N}_{\mathcal{T}} : i \in \mathcal{N}^k\}$, $\mathcal{A}_{\mathcal{T}}^k = \{((i, t), (j, t')) \in \mathcal{A}_{\mathcal{T}} : (i, j) \in \mathcal{A}^k\}$, and $\mathcal{H}_{\mathcal{T}}^k = \{((i, t), (i, t+1)) \in \mathcal{H}_{\mathcal{T}} : i \in \mathcal{N}^k\}$ denote the portions of the time-expanded network that can be used by commodity k .

With this notation, we let the binary decision variable $x_{ij}^{ktt'}$ indicate whether commodity $k \in \mathcal{K}$ uses arc $((i, t), (j, t')) \in \mathcal{A}_{\mathcal{T}}^k \cup \mathcal{H}_{\mathcal{T}}^k$. We let the integer decision variable $y_{ij}^{tt'}$ indicate the number of vehicles that move on arc $((i, t), (j, t')) \in \mathcal{A}_{\mathcal{T}}^k$. The time-expanded network formulation of the SNSP based on the network $\mathcal{D}_{\mathcal{T}}$, that we refer to as *TEN-SNSP*($\mathcal{D}_{\mathcal{T}}$), follows.

$$z_{TEN}(\mathcal{D}_{\mathcal{T}}) = \text{minimize} \quad \sum_{((i, t), (j, t')) \in \mathcal{A}_{\mathcal{T}}} f_{ij} y_{ij}^{tt'} \quad (1)$$

subject to

$$\sum_{((i, t), (j, t')) \in \mathcal{A}_{\mathcal{T}}^k \cup \mathcal{H}_{\mathcal{T}}^k} x_{ij}^{ktt'} - \sum_{((j, t'), (i, t)) \in \mathcal{A}_{\mathcal{T}}^k \cup \mathcal{H}_{\mathcal{T}}^k} x_{ji}^{kt't} = \begin{cases} 1 & \text{if } (i, t) = (o_k, e_k) \\ -1 & \text{if } (i, t) = (d_k, l_k) \\ 0 & \text{otherwise.} \end{cases} \quad \forall k \in \mathcal{K}, \forall (i, t) \in \mathcal{N}_{\mathcal{T}}^k \quad (2)$$

$$\sum_{k \in \mathcal{K} : ((i, t), (j, t')) \in \mathcal{A}_{\mathcal{T}}^k} q_k x_{ij}^{ktt'} \leq u_{ij} y_{ij}^{tt'} \quad \forall ((i, t), (j, t')) \in \mathcal{A}_{\mathcal{T}}, \quad (3)$$

$$x_{ij}^{ktt'} \in \{0, 1\} \quad \forall ((i, t), (j, t')) \in \mathcal{A}_{\mathcal{T}}^k \cup \mathcal{H}_{\mathcal{T}}^k, k \in \mathcal{K}, \quad (4)$$

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

The objective (1) seeks to minimize the total costs associated with vehicle movements. Constraints (2) are standard flow conservation constraints, ensuring that each commodity departs its origin terminal at its earliest available time and arrives at its destination terminal at its latest due time. Constraints (3) ensure sufficient vehicle capacity is scheduled to support shipment movements. Constraints (4) and (5) define the decision variables and their domains. We note that the existence of a single path, p_k , on which commodity k may be routed is reflected in the domains defined in constraints (4).

One of the primary computational challenges associated with solving such a formulation is the size of the complete time-expanded network, $\mathcal{D}_{\mathcal{T}}$, and the resulting number of decision variables in the model. However, similar to what was shown in Proposition 1 of Boland et al. (2017a) for the Scheduled Service Network Design Problem, a reduced time-expanded network $\mathcal{D}_{\mathcal{T}}^R$ that consists of a well-chosen subset of points $(i, t) \in \mathcal{N}_{\mathcal{T}}$ can be formed such that the optimal solution of the formulation $TEN-SNSP(\mathcal{D}_{\mathcal{T}}^R)$ is also an optimal solution to $TEN-SNSP(\mathcal{D}_{\mathcal{T}})$. We restate this Proposition, albeit stated in terms of the SNSP below.

Proposition 1. *Consider a time-expanded network $\mathcal{D}_{\mathcal{T}}^R$ based on a set $\mathcal{N}_{\mathcal{T}}^R$ that contains nodes of the form (o_k, e_k) or $(v_j^k, e_k + \sum_{i=1}^{j-1} \tau_{v_i^k v_{i+1}^k})$, $j = 2, \dots, r_k$, $\forall k \in K$ and arc sets $\mathcal{A}_{\mathcal{T}}^R, \mathcal{H}_{\mathcal{T}}^R$ constructed accordingly. An optimal solution to the $TEN-SNSP(\mathcal{D}_{\mathcal{T}})$ can be determined by formulating and solving $TEN-SNSP(\mathcal{D}_{\mathcal{T}}^R)$.*

Proposition 1 states that if the network is defined for a subset of nodes that represent terminals at times that are the earliest at which some commodity could reach that terminal, then it is sufficient to find the optimal solution to the SNSP.

We can also strengthen the formulation $TEN-SNSP(\mathcal{D}_{\mathcal{T}})$ with valid inequalities similar to the *Arc-Time-Window* inequalities presented in Hewitt (2019). To define these inequalities, we let $q_{ij} = \sum_{k \in \mathcal{K}_{ij}} q_k$ denote the total size of all commodities whose path includes arc (i, j) . Then, we must have

$$\sum_{t, t' \in \mathcal{T}: ((i, t), (j, t')) \in \mathcal{A}_{\mathcal{T}}} y_{ij}^{tt'} \geq \lceil \frac{q_{ij}}{u_{ij}} \rceil \quad \forall (i, j) \in \mathcal{A}. \quad (6)$$

We refer to these inequalities as *Agg-Arc-Capacity* inequalities.

Proposition 1 enables the $TEN-SNSP(\mathcal{D}_{\mathcal{T}})$ to be formulated in a static manner, albeit with a reduced node set. That formulation can then be strengthened with the *Agg-Arc-Capacity* inequalities. However, it is fairly easy to see that the *Dynamic Discretization Discovery* algorithmic framework presented in Boland et al. (2017a), and enhanced in Hewitt (2019), can be applied with nearly no modification to solve the SNSP.

3.3 Consolidation-based formulation

We next present a formulation of the SNSP that is based on sets of possible consolidations on each arc, $(i, j) \in \mathcal{A}$. After presenting the formulation, we prove its equivalence to $TEN-SNSP(\mathcal{D}_{\mathcal{T}})$.

To define the formulation we let $\mathcal{C}_{ij} = \{C^1, \dots, C^{m_{ij}}\}$, $C^g \subseteq \mathcal{K}_{ij} \quad \forall g = 1, \dots, m_{ij}$, denote all sets of commodities that can dispatch on arc (i, j) at the same time. Recalling that $[\alpha_i^k, \beta_i^k]$ denotes the time window during which commodity k must depart from node i , we have that $C^g = \{k_1^g, \dots, k_{m_g}^g\} \in \mathcal{C}_{ij} \quad \forall g = 1, \dots, m_{ij}$ if and only if $\cap_{g=1}^{m_g} [\alpha_i^{k_g}, \beta_i^{k_g}] \neq \emptyset$. In words, a consolidation is feasible on an arc if and only if the time windows for all commodities in that consolidation overlap in at least one time point. That said, \mathcal{C}_{ij} also contains all singleton sets. Namely, if $k \in \mathcal{K}_{ij}$, then there exists $g \in [1, m_{ij}]$ such that $C^g = \{k\}$. In addition, we let $\mathcal{C} = \cup_{(i, j) \in \mathcal{A}} \mathcal{C}_{ij}$. Throughout this paper we will refer to “consolidation C^g ” as shorthand for the set of commodities $\{k_1^g, \dots, k_{m_g}^g\}$ contained in that consolidation. Regarding data elements associated with these consolidation sets, we let the attribute $\phi_C^k \in \{0, 1\}$, $k \in \mathcal{K}, C \in \mathcal{C}$ represent whether $k \in C$. We also let $s_C = \lceil \frac{\sum_{k \in C} q_k}{u_{ij}} \rceil$ represent the number of vehicles needed on arc $(i, j) \in \mathcal{A}$ to transport consolidation $C \in \mathcal{C}_{ij}$.

To model the SNSP, we let the binary variable w_C indicate whether the consolidation consisting of commodities in set $C \in \mathcal{C}_{ij}$ traveling together on arc $(i, j) \in \mathcal{A}$ is chosen. Note this choice implies that the commodities must dispatch at the same time. We let the integer variable y_{ij} represent the number of vehicles that dispatch on arc $(i, j) \in \mathcal{A}$. We let the decision variables $\gamma_{v_i^k v_{i+1}^k}^k, i = 1, \dots, r_k - 1$ prescribe the time at which commodity k dispatches on the arc $(v_i^k, v_{i+1}^k), i = 1, \dots, r_k - 1$ on its path. With these sets and decision variables, we define the optimization problem $Cons-SNSP(\mathcal{C})$, as

$$z_{Cons}(\mathcal{C}) = \text{minimize} \sum_{(i,j) \in \mathcal{A}} f_{ij} y_{ij} \quad (7)$$

subject to

$$\sum_{C \in \mathcal{C}_{ij}} \phi_C^k w_C = 1 \quad \forall k \in \mathcal{K}, (i,j) \in \mathcal{A}^k, \quad (8)$$

$$\sum_{C \in \mathcal{C}_{ij}} s_C w_C \leq y_{ij} \quad \forall (i,j) \in \mathcal{A}, \quad (9)$$

$$\gamma_{ij}^k - \gamma_{ij}^{k'} \leq M_i^{kk'} (1 - \sum_{C \in \mathcal{C}_{ij}} \phi_C^k \phi_C^{k'} w_C) \quad \forall (i,j) \in \mathcal{A}, k, k' \in \mathcal{K}_{ij}, \quad (10)$$

$$\gamma_{v_i^k v_{i+1}^k}^k + \tau_{v_i^k v_{i+1}^k} \leq \gamma_{v_{i+1}^k v_{i+2}^k}^k \quad \forall k \in \mathcal{K}, i = 1, \dots, r_k - 2, \quad (11)$$

$$\alpha_{v_i^k} \leq \gamma_{v_i^k v_{i+1}^k}^k \leq \beta_{v_i^k} \quad \forall k \in \mathcal{K}, i = 1, \dots, r_k - 1, \quad (12)$$

$$\gamma_{v_i^k v_{i+1}^k}^k \in \mathbb{N} \quad \forall (v_i^k, v_{i+1}^k) \in p_k, k \in \mathcal{K}, \quad (13)$$

$$w_C \in \{0, 1\} \quad \forall C \in \mathcal{C}, \quad (14)$$

$$y_{ij} \in \mathbb{N} \quad \forall (i,j) \in \mathcal{A}. \quad (15)$$

The objective seeks to minimize the total costs associated with vehicle moves that transport consolidations of shipments. Constraints (8) ensure that a consolidation is chosen for each commodity on each arc in its path. Constraints (9) ensure that sufficient capacity is paid for on each arc to support the consolidations chosen for that arc. Constraints (10) ensure that all commodities in the consolidation chosen for an arc dispatch at the same time. Constraints (11) ensure that the dispatch times for arcs on the path of a commodity agree with their travel times. Constraints (12) ensure the dispatch time decision variables occur within the corresponding time windows. Constraints (13), (14), and (15) define the decision variables and their domains.

While the big-M value in constraints (10) may lead to a weak formulation, the formulation can be tightened without rendering any solutions infeasible by setting $M_i^{kk'} = \beta_i^k - \alpha_i^{k'}$. Lastly, we note that we can strengthen the *Cons-SNSP*(\mathcal{C}) formulation with the inequality $y_{ij} \geq \lceil \frac{q_{ij}}{u_{ij}} \rceil, \forall (i,j) \in \mathcal{A}$, which is analogous to the *Agg-Arc-Capacity* inequality for the *TEN-SNSP*($\mathcal{D}_{\mathcal{T}}$).

Solving the proposed consolidation-based formulation will yield the same optimal objective function value as the time-expanded network-based formulation. More precisely, we have the following theorem. Its proof can be found in Appendix A.

Theorem 1. $z_{TEN}(\mathcal{D}_{\mathcal{T}}) = z_{Cons}(\mathcal{C})$.

We note that if all travel times, τ , and commodity early, e , and late times, l , are integral the constraints (13) can be relaxed to allow the γ variables to take on fractional values. This can be seen by first observing that any solution with fractional values for some γ variables can be converted to another feasible solution of *Cons-SNSP*(\mathcal{C}) with integral values for all γ variables through rounding. Second, as the γ variables do not appear in the objective, this rounding does not change the objective function value of the solution.

Lastly, we provide an example of the data derived to formulate the *Cons-SNSP*(\mathcal{C}).

Example 1. Figure 1 illustrates a network, \mathcal{N} , with four nodes and three arcs. All arcs have a travel time of one. Three commodities are to be routed through this network. Solid arrows in the figure indicate arcs in \mathcal{A} , while dashed arrows indicate the origin and destination terminals for each commodity. Table 1 presents data regarding each commodity, including its origin and destination terminals, its early and late times, its path, its size quoted as a fraction of vehicle capacity, and its time window at each terminal on its path. The time windows identified in Table 1 for each commodity at each of the nodes a, b , and c (if applicable) are derived from that commodity's early and late times, knowledge of the path it must take, and travel times on arcs in that path.

To construct the consolidation-based formulation, we enumerate all possible consolidations on arcs $(a,b), (b,c)$, and (c,d) given the time windows at nodes a, b , and c identified in Table 1. In addition to displaying them in Figure 1 below each arc, we list them in Table 2 along with other attributes of each consolidation.

The *SNSP* formulation for this example is detailed in Appendix B.

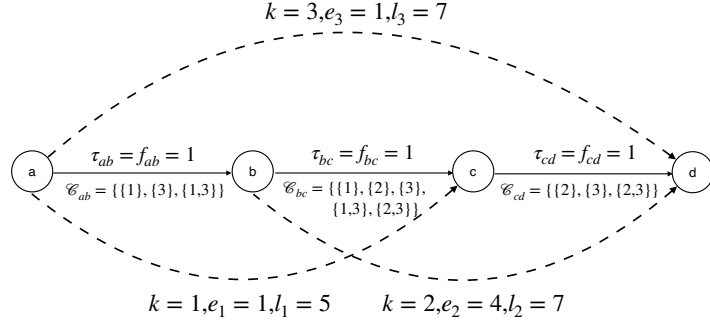


Figure 1: Network \mathcal{N} and commodities \mathcal{K} in Example 1

Table 1: Commodity attributes in Example 1

k	o_k	d_k	e_k	l_k	p_k	q_k	Time windows		
							$[\alpha_a^k, \beta_a^k]$	$[\alpha_b^k, \beta_b^k]$	$[\alpha_c^k, \beta_c^k]$
1	a	c	1	4	a \rightarrow b \rightarrow c	0.75	1,2	2,3	-
2	b	d	4	7	b \rightarrow c \rightarrow d	0.55	-	4,5	5,6
3	a	d	1	7	a \rightarrow b \rightarrow c \rightarrow d	0.65	1,4	2,5	3,6

Table 2: Consolidations \mathcal{C}

Arc	\mathcal{C}	$\sum_{k \in C_{ij}} q_k$	$s_{C_{ij}}$
a \rightarrow b	$C_{ab}^1 = \{1\}$	0.75	1
a \rightarrow b	$C_{ab}^2 = \{3\}$	0.65	1
a \rightarrow b	$C_{ab}^3 = \{1, 3\}$	1.40	2
b \rightarrow c	$C_{bc}^1 = \{1\}$	0.75	1
b \rightarrow c	$C_{bc}^2 = \{2\}$	0.55	1
b \rightarrow c	$C_{bc}^3 = \{3\}$	0.65	1
b \rightarrow c	$C_{bc}^4 = \{1, 3\}$	1.40	2
b \rightarrow c	$C_{bc}^5 = \{2, 3\}$	1.20	2
c \rightarrow d	$C_{cd}^1 = \{2\}$	0.55	1
c \rightarrow d	$C_{cd}^2 = \{3\}$	0.65	1
c \rightarrow d	$C_{cd}^3 = \{2, 3\}$	1.20	2

3.4 Analytical comparison of formulations

We next compare the linear relaxations of each formulation. More specifically, we prove in Theorem 2 that solving the linear relaxation of $TEN\text{-}SNSP(\mathcal{D}_{\mathcal{T}})$ can not yield a larger bound than solving the linear relaxation of $Cons\text{-}SNSP(\mathcal{C})$.

Theorem 2. *Given the definitions of $\mathcal{D}_{\mathcal{T}}$ and \mathcal{C} in Theorem 1, let $LP\text{-}TEN$ and $LP\text{-}Cons$ be the linear relaxations of problems $TEN\text{-}SNSP(\mathcal{D}_{\mathcal{T}})$ and $Cons\text{-}SNSP(\mathcal{C})$. Let $z_{LP\text{-}TEN}^*$ and $z_{LP\text{-}Cons}^*$ be the objective function values of optimal solutions for each problem. Then $z_{LP\text{-}TEN}^* \leq z_{LP\text{-}Cons}^*$.*

Proof. We first consider $LP\text{-}TEN$ and an optimal solution (x^*, y^*) to that linear program. As $LP\text{-}TEN$ is a minimization problem, in the absence of (6), the objective of $LP\text{-}TEN$ implies that $y_{ij}^{*tt'} = \frac{\sum_{k \in \mathcal{K}: ((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}}^k} q_k x_{ij}^{*ktt'}}{u_{ij}}$.

Relatedly, for a given $(i, j) \in \mathcal{A}$ we have that $\sum_{((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}}^k} y_{ij}^{*tt'} = \sum_{((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}}^k} \frac{\sum_{k \in \mathcal{K}: ((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}}^k} q_k x_{ij}^{*ktt'}}{u_{ij}}$
 $= \sum_{k \in \mathcal{K}: ((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}}^k} \frac{q_k}{u_{ij}} \sum_{((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}}^k} x_{ij}^{*ktt'} = \sum_{k \in \mathcal{K}: ((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}}^k} \frac{q_k}{u_{ij}}$. The last equality is due to the flow balance constraints (2) and the presumption that (i, j) is in the path already specified for each commodity $k \in \mathcal{K}_{ij}$. As $\lceil x \rceil \geq x \quad \forall x \in \mathbb{R}$, when (6) is included in the formulation we have $\sum_{((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}}^k} y_{ij}^{*tt'} = \lceil \frac{q_{ij}}{u_{ij}} \rceil$. Thus, we have

$$z_{LP\text{-}TEN}^* = \sum_{(i,j) \in \mathcal{A}} f_{ij} \lceil \frac{q_{ij}}{u_{ij}} \rceil. \quad (16)$$

Next, we consider $LP\text{-}Cons$, a single arc $(i, j) \in \mathcal{A}$ such that $\mathcal{K}_{ij} \neq \emptyset$, and the constraints $\sum_{C \in \mathcal{C}_{ij}} \phi_C^k w_C = 1, \forall k \in \mathcal{K}_{ij}$ (constraints (8) in the formulation). Summing constraints (8) over $k \in \mathcal{K}_{ij}$ we have $\sum_{k \in \mathcal{K}_{ij}} \sum_{C \in \mathcal{C}_{ij}} \phi_C^k w_C = \sum_{C \in \mathcal{C}_{ij}} |C| w_C = |\mathcal{K}_{ij}|$. As $|C| \leq |\mathcal{K}_{ij}|$ we have $\sum_{C \in \mathcal{C}_{ij}} \frac{|C|}{|\mathcal{K}_{ij}|} w_C = 1$ and thus $\sum_{C \in \mathcal{C}_{ij}} w_C \geq 1$. From constraints (9) we have that $y_{ij} \geq \sum_{C \in \mathcal{C}_{ij}} s_C w_C = \sum_{C \in \mathcal{C}_{ij}} \lceil \frac{\sum_{k \in C} q_k}{u_{ij}} \rceil w_C$. Thus, as $\lceil x \rceil + \lceil y \rceil \geq \lceil x + y \rceil$ and $\sum_{C \in \mathcal{C}_{ij}} w_C \geq 1$ we have that $y_{ij} \geq \lceil \frac{q_{ij}}{u_{ij}} \rceil$. As such, all solutions to $LP\text{-}Cons$ satisfy $y_{ij} \geq \lceil \frac{q_{ij}}{u_{ij}} \rceil$ and thus $z_{LP\text{-}Cons}^* \geq \sum_{(i,j) \in \mathcal{A}} f_{ij} \lceil \frac{q_{ij}}{u_{ij}} \rceil = z_{LP\text{-}TEN}^*$. \square

In addition, there are also instances wherein $z_{LP\text{-}Cons}^* > z_{LP\text{-}TEN}^*$. For example, consider an instance of the $LP\text{-}TEN$ derived from Example 1. In addition to constraints (6), which reflect the vehicle capacity needed when all three commodities dispatch at the same time, one can examine the time windows at b for commodities 1 and 3 to derive that $y_{bc}^{23} + y_{bc}^{34} \geq 1$ and $y_{bc}^{45} + y_{bc}^{56} \geq 1$.

We illustrate in Figure 2a an optimal solution to this linear program, including these two additional inequalities. Recalling that in this example $f_{ij} = 1, \forall (i, j) \in \mathcal{A}$, we see that $z_{LP\text{-}TEN}^* = 6$. We illustrate an optimal solution to the corresponding formulation of the $LP\text{-}Cons$ on a time-expanded network in Figure 2b. We first note that the w variables naturally take on binary values in this solution. In addition, as the s_C coefficients associated with these variables are derived from applying the $\lceil \cdot \rceil$ operator, the solution to this linear program better approximates the vehicle capacity needed. More precisely, we have $y_{ab} = 2, y_{bc} = 3$, and $y_{cd} = 2$. Thus, we have $z_{LP\text{-}Cons}^* = 7 > 6 = z_{LP\text{-}TEN}^*$. In addition, as the values of all decision variables in the solution to $LP\text{-}Cons$ fall within the domains required by the $Cons\text{-}SNSP(\mathcal{C})$, the solution is also the optimal solution to $Cons\text{-}SNSP(\mathcal{C})$.

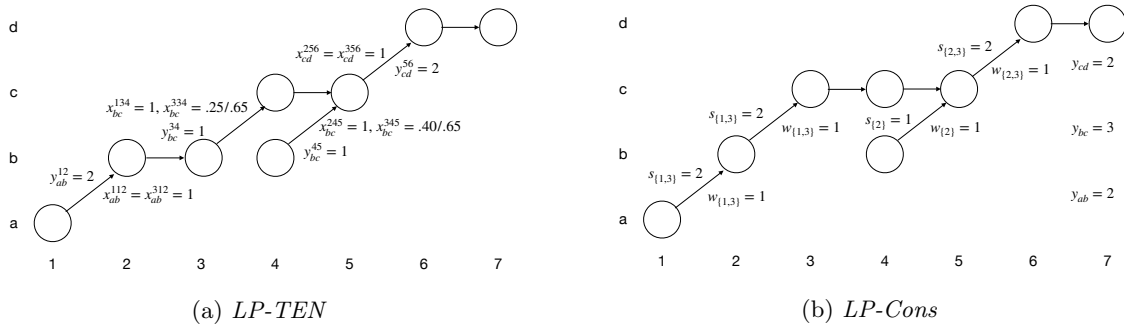


Figure 2: Optimal linear relaxation solutions for Example 1

4 Integer programming-based column generation

The analysis in Section 3.4 suggests that $Cons-SNSP(\mathcal{C})$ is the stronger formulation of the two proposed in Section 3 for the SNSP. However, its enumerative nature of is likely to lead to computational challenges when solving larger instances of the formulation. As such, in this section we present an algorithm for solving the $Cons-SNSP(\mathcal{C})$ that does not enumerate the set \mathcal{C} *a priori*, but instead dynamically in the course of its execution. In this section, we present an algorithm that at each iteration solves a relaxation of the original problem and then uses information from the solution to that relaxation to determine variables to add to the relaxation.

In this section, we first present the relaxation the algorithm solves. We then illustrate with an example why it is a relaxation as well as formally prove this claim. We then outline the reasons why solutions to that relaxation may not be feasible for the $Cons-SNSP(\mathcal{C})$ and how the algorithm chooses to strengthen the relaxation to render such solutions infeasible. We finish with a formal description of the algorithm and a proof of its correctness.

4.1 Relaxation of $Cons-SNSP(\mathcal{C})$

In this section, we present an integer program that is similar in form to $Cons-SNSP(\mathcal{C})$, but defined over only a subset of \mathcal{C} . Yet, the integer program is a relaxation of $Cons-SNSP(\mathcal{C})$. We conclude this section with a proof of this claim.

This formulation is defined on subsets, $\underline{\mathcal{C}}_{ij} = \{\underline{\mathcal{C}}^1, \dots, \underline{\mathcal{C}}^{n_{ij}}\} \subseteq \mathcal{C}_{ij}, n_{ij} \leq n_{ij}$ of the potential consolidations for each arc $(i, j) \in \mathcal{A}$. We note that while $\underline{\mathcal{C}}^g \in \underline{\mathcal{C}}_{ij}$, it is not necessarily the case that $\underline{\mathcal{C}}^g = \mathcal{C}^g$, wherein $\mathcal{C}^g \in \mathcal{C}_{ij}$. As in the presentation of $Cons-SNSP(\mathcal{C})$, we let $\underline{\mathcal{C}} = \cup_{(i,j) \in \mathcal{A}} \underline{\mathcal{C}}_{ij}$. The formulation is defined with the same y_{ij} and $\gamma_{v_i^k v_{i+1}^k}$ variables as in $Cons-SNSP(\mathcal{C})$. The binary variables $w_{\underline{\mathcal{C}}}$ are defined as above, albeit only over the consolidations in sets $\underline{\mathcal{C}}_{ij}$. The values $\phi_{\underline{\mathcal{C}}}^k$ are also defined as above.

However, the values $\underline{s}_{\underline{\mathcal{C}}}$ are defined differently from the values $s_{\mathcal{C}}$ used to formulate the $Cons-SNSP(\mathcal{C})$. Specifically, they are defined as

$$\underline{s}_{\underline{\mathcal{C}}} = \begin{cases} s_{\underline{\mathcal{C}}} & \text{if } \forall \mathcal{C}' \subseteq \underline{\mathcal{C}}, \mathcal{C}' \in \underline{\mathcal{C}}_{ij}, \\ \min_{\mathcal{C}' \subseteq \underline{\mathcal{C}}: \mathcal{C}' \notin \underline{\mathcal{C}}_{ij}} s_{\mathcal{C}'} & \text{o.w.} \end{cases} \quad (17)$$

The if portion of equation (17) tests whether every subset of $\underline{\mathcal{C}}$ is an element of $\underline{\mathcal{C}}_{ij}$. In this case, the vehicle capacity associated with $\underline{\mathcal{C}}$ in $Rel-Cons-SNSP(\underline{\mathcal{C}})$ is the actual capacity needed to transport the commodities of $\underline{\mathcal{C}}$. When this condition is not satisfied, $\underline{s}_{\underline{\mathcal{C}}}$ is set to require the least vehicle capacity required of any subset of $\underline{\mathcal{C}}$ that is *not* an element of $\underline{\mathcal{C}}_{ij}$. As a result, the calculation of $\underline{s}_{\underline{\mathcal{C}}}$ ensures that the number of vehicles needed to transport consolidation $\underline{\mathcal{C}}$ is no greater than the number needed by any subset of $\underline{\mathcal{C}}$ that is not currently considered by the relaxation. Note, (17) implies that $\underline{s}_{\underline{\mathcal{C}}} \leq s_{\underline{\mathcal{C}}}$.

This definition is illustrated on the following example.

Example 2. We focus on a single arc (i, j) and four commodities with a path that contains that arc (e.g. $\mathcal{K}_{ij} = \{1, 2, 3, 4\}$). Data for this example is provided in Tables 3, 4, and 5. Table 3 lists each commodity $k \in \mathcal{K}_{ij}$ and its size, q_k . Table 4 lists each of the possible consolidations of those commodities on arc (i, j) , along with its total size and the number of vehicles it requires. We note that the set of maximal consolidations is $\mathcal{C}_{ij}^{max} = \{\{1, 2, 3\}, \{1, 3, 4\}, \{1, 4\}, \{3, 4\}\}$. Table 5 provides a subset of these consolidations that can be used when formulating $Rel-Cons-SNSP(\underline{\mathcal{C}})$. We note that each of the maximal consolidations in Table 4 is listed in Table 5.

With these sets and decision variables we define the following optimization problem, which we refer to as $Rel-Cons-SNSP(\underline{\mathcal{C}})$.

$$z_{Rel-Cons}(\underline{\mathcal{C}}) = \text{minimize} \sum_{(i,j) \in \mathcal{A}} f_{ij} y_{ij}$$

Table 3: Commodities \mathcal{K}_{ij} and their sizes

Commodity k	q_k
1	0.80
2	0.20
3	0.20
4	0.20

Table 4: Possible consolidations, \mathcal{C}_{ij}

C	$\sum_{k \in C} q_k$	s_C
1,2,3	1.20	2.00
1,3,4	1.20	2.00
1,2	1.00	1.00
1,3	1.00	1.00
1,4	1.00	1.00
2,3	0.40	1.00
3,4	0.40	1.00
1	1.00	1.00
2	1.00	1.00
3	1.00	1.00
4	1.00	1.00

Table 5: Subset of consolidations, \mathcal{C}_{ij}

\underline{C}	\underline{s}_C
1,2,3	1
1,3,4	1
1,4	1
3,4	1
1	1
2	1
3	1
4	1

subject to

$$\sum_{\underline{C} \in \mathcal{C}_{ij}} \phi_{\underline{C}}^k w_{\underline{C}} \geq 1 \quad \forall k \in \mathcal{K}, (i, j) \in \mathcal{A}^k, \quad (18)$$

$$\sum_{\underline{C} \in \mathcal{C}_{ij}} \underline{s}_{\underline{C}} w_{\underline{C}} \leq y_{ij} \quad \forall (i, j) \in \mathcal{A}, \quad (19)$$

$$\gamma_{ij}^k - \gamma_{ij}^{k'} \leq M(1 - \phi_{\underline{C}}^k \phi_{\underline{C}'}^{k'} w_{\underline{C}}) \quad \forall (i, j) \in \mathcal{A}, \underline{C} \in \mathcal{C}_{ij}, k, k' \in \mathcal{K}_{ij} \text{ such that } \underline{C} \setminus \{k\}, \underline{C}' \setminus \{k'\} \in \mathcal{C}_{ij}, \quad (20)$$

$$\gamma_{v_i^k v_{i+1}^k}^k + \tau_{v_i^k v_{i+1}^k} \leq \gamma_{v_{i+1}^k v_{i+2}^k}^k \quad \forall k \in \mathcal{K}, i = 1, \dots, r_k - 2, \quad (21)$$

$$\alpha_{v_i^k} \leq \gamma_{v_i^k v_{i+1}^k}^k \leq \beta_{v_i^k} \quad \forall k \in \mathcal{K}, i = 1, \dots, r_k, \quad (22)$$

$$\gamma_{v_i^k v_{i+1}^k}^k \in \mathbb{N} \quad \forall (v_i^k, v_{i+1}^k) \in p_k, k \in \mathcal{K}, \quad (23)$$

$$w_{\underline{C}} \in \{0, 1\} \quad \forall (i, j) \in \mathcal{A}, \underline{C} \in \mathcal{C}_{ij}, \quad (24)$$

$$y_{ij} \in \mathbb{N} \quad \forall (i, j) \in \mathcal{A} \quad (25)$$

Many of the constraints defining $Rel\text{-}Cons\text{-}SNSP(\underline{\mathcal{C}})$ are the same as those defining $Cons\text{-}SNSP(\mathcal{C})$. The constraints that differ are constraints (18), (19), and (20). Constraints (18) ensure a commodity appears in at least one chosen consolidation for each arc in its path. Constraints (19) are of the same form as constraints (9) of $Cons\text{-}SNSP(\mathcal{C})$ only with the coefficients $\underline{s}_{\underline{C}}$. Constraints (20) ensure that two commodities dispatch on an arc at the same time under two conditions: (1) a consolidation is chosen for that arc that contains those two commodities, and, (2) for each commodity the set \underline{C} also contains that consolidation without that commodity. The first condition is similar in spirit to what defines constraints (10) of $Cons\text{-}SNSP(\mathcal{C})$. The second plays a role in ensuring $Rel\text{-}Cons\text{-}SNSP(\underline{\mathcal{C}})$ is a relaxation, as we will next see.

When $Rel\text{-}Cons\text{-}SNSP(\underline{\mathcal{C}})$ is defined with the appropriate set $\underline{\mathcal{C}}$ (e.g. using the algorithms proposed in Section 4.2), it can be shown to be a relaxation of $Cons\text{-}SNSP(\mathcal{C})$. To be more precise, we let \mathcal{C}_{ij}^{max} denote the set of maximal consolidations on arc $(i, j) \in \mathcal{A}$. Formally, the consolidation $C' \in \mathcal{C}_{ij}^{max}$ if and only if there does not exist $C'' \in \mathcal{C}_{ij}$ and $C' \subset C''$. We later prove that $Rel\text{-}Cons\text{-}SNSP(\underline{\mathcal{C}})$ is a relaxation when formulated with a consolidation set for each arc that contains all the maximal consolidations for that arc. More precisely, when formulated with sets $\underline{\mathcal{C}}_{ij}$ such that $\forall C \in \mathcal{C}_{ij}^{max}, C \in \underline{\mathcal{C}}_{ij}$. The proof of this statement involves showing that any solution to $Cons\text{-}SNSP(\mathcal{C})$ can be used to construct a solution to the $Rel\text{-}Cons\text{-}SNSP(\underline{\mathcal{C}})$ of equal objective function value. However, before formally proving this result we illustrate why $Rel\text{-}Cons\text{-}SNSP(\underline{\mathcal{C}})$ is a relaxation in the case of Example 2.

Example 3. (continued from Example 2) Consider a solution $(\bar{w}, \bar{\gamma}, \bar{y})$ to $Cons\text{-}SNSP(\mathcal{C})$ that involves the consolidations $\{1, 2\}$ and $\{3, 4\}$. Namely, $\bar{w}_{\{1,2\}} = \bar{w}_{\{3,4\}} = 1$. This in turn requires $\bar{\gamma}_{ij}^1 = \bar{\gamma}_{ij}^2$ and $\bar{\gamma}_{ij}^3 = \bar{\gamma}_{ij}^4$. Suppose that $\bar{\gamma}_{ij}^1 = \bar{\gamma}_{ij}^2 = 1$ and $\bar{\gamma}_{ij}^3 = \bar{\gamma}_{ij}^4 = 3$. Collectively, these two consolidations require two vehicles to travel on arc (i, j) (i.e. $\bar{y}_{ij} = 2$). We will use this solution to construct a solution $(\underline{w}, \underline{\gamma}, \underline{y})$ to the $Rel\text{-}Cons\text{-}SNSP(\underline{\mathcal{C}})$ of equal objective function value.

We start by setting $y_{ij} = \bar{y}_{ij} = 2$. Thus, the constructed solution uses the same amount of vehicle capacity and the objective function values of the solution to the $\text{Cons-SNSP}(\mathcal{C})$ and the constructed solution will be the same. We next set $\gamma_{ij}^k = \bar{\gamma}_{ij}^k, k = 1, 2, 3, 4$. In words, commodities dispatch at the same times in the two solutions.

Regarding the \underline{w} variables, for each consolidation chosen in the solution to the $\text{Cons-SNSP}(\mathcal{C})$ we identify a corresponding consolidation to be chosen in the solution to the $\text{Rel-Cons-SNSP}(\underline{\mathcal{C}})$. First, we consider consolidation $\{3, 4\}$, as $\bar{w}_{\{3,4\}} = 1$. As $\{3, 4\} \in \underline{\mathcal{C}}_{ij}$, we set $\underline{w}_{\{3,4\}} = 1$. We next consider consolidation $\{1, 2\}$, as $\bar{w}_{\{1,2\}} = 1$. Here, we observe that $\{1, 2\} \notin \underline{\mathcal{C}}_{ij}$ and thus we identify the minimal consolidation in $\underline{\mathcal{C}}_{ij}$ that contains $\{1, 2\}$. That such a consolidation exists is guaranteed as $\underline{\mathcal{C}}_{ij}$ contains the maximal consolidations $\{1, 2, 3\}, \{1, 3, 4\}, \{1, 4\}, \{3, 4\}$. In this case, $\{1, 2, 3\}$ is the minimal consolidation containing $\{1, 2\}$ and we set $\underline{w}_{\{1,2,3\}} = 1$.

We next discuss why this solution is feasible for the $\text{Rel-Cons-SNSP}(\underline{\mathcal{C}})$. We note that while commodity 3 appears in two chosen consolidations, constraints (18) are defined in such a manner that this is feasible. Next, we note that given the sizes of the commodities in the consolidations $\{1, 2, 3\}$ and $\{3, 4\}$ they collectively require three vehicles ($s_{\{1,2,3\}} = 2, s_{\{3,4\}} = 1$). However, because the consolidation $\{1, 2\}$ is not present in $\underline{\mathcal{C}}_{ij}$, $\text{Rel-Cons-SNSP}(\underline{\mathcal{C}})$ is formulated with $\underline{s}_{\{1,2,3\}} = 1$.

Thus, for constraints (19) we have $\underline{s}_{\{1,2,3\}}\underline{w}_{\{1,2,3\}} + \underline{s}_{\{3,4\}}\underline{w}_{\{3,4\}} = 1 + 1 = 2 = \underline{y}_{ij}$ and the constraint is satisfied.

We next consider constraints (20). Note these constraints are defined for a pair of commodities k, k' and a given consolidation \underline{C} only if the relaxation considers the consolidations $\underline{C} \setminus \{k\}$ and $\underline{C} \setminus \{k'\}$ (i.e. they are in $\underline{\mathcal{C}}_{ij}$). Regarding the chosen consolidation $\{3, 4\}$ we note that as $\underline{\mathcal{C}}_{ij}$ contains the consolidations $\{3\}$ and $\{4\}$ then the constraint is defined for commodities 3 and 4 and consolidation $\{3, 4\}$. Given that $\gamma_{ij}^3 = \bar{\gamma}_{ij}^3 = \bar{\gamma}_{ij}^4 = \gamma_{ij}^4$, the constraint is satisfied.

While $\underline{w}_{\{1,2,3\}} = 1$, as consolidation $\{1, 2\}$ is not contained in $\underline{\mathcal{C}}_{ij}$, constraint (20) is not defined for commodities 2 and 3. Thus, $\underline{w}_{\{1,2,3\}}$ can take on the value 1 even though $\gamma_{ij}^3 \neq \gamma_{ij}^2$. Like vehicle capacity, constraints (20) are defined based on the recognition that consolidation $\{1, 2, 3\}$ may serve as a proxy for consolidations it contains that are not represented in $\underline{\mathcal{C}}$. When it does, not all commodities in that consolidation are required to dispatch at the same time.

Lastly, we note that as the commodities dispatch at the same time in the two solutions, the $\underline{\gamma}$ variables satisfy constraints (21), (22), and (23) of the $\text{Rel-Cons-SNSP}(\underline{\mathcal{C}})$ because the $\bar{\gamma}$ variables satisfy the analogous constraints in the $\text{Cons-SNSP}(\mathcal{C})$. Similarly, the domain constraints (24) and (25) are clearly satisfied.

This example illustrates the logic of the proof of the following result.

Theorem 3. Consider the directed network \mathcal{D} and commodity set \mathcal{K} . Let $\underline{\mathcal{C}}_{ij} = \{\underline{C}^1, \dots, \underline{C}^{n_{ij}}\} \subseteq \mathcal{C}_{ij}, n_{ij} \leq n_{ij}$ and $\underline{\mathcal{C}} = \cup_{(i,j) \in \mathcal{A}} \underline{\mathcal{C}}_{ij}$. If for each arc $(i, j) \in \mathcal{A}$ the set $\underline{\mathcal{C}}_{ij}$ contains all maximal consolidations (e.g. $\forall C \in \mathcal{C}_{ij}^{max}, C \in \underline{\mathcal{C}}_{ij}$), then $z_{\text{Rel-Cons}}(\underline{\mathcal{C}}) \leq z_{\text{Cons}}(\mathcal{C})$.

Proof. Consider a feasible solution $(\bar{\gamma}, \bar{w}, \bar{y})$ of $\text{Cons-SNSP}(\mathcal{C})$. We will show that this solution can be used to construct a feasible solution $(\underline{\gamma}, \underline{w}, \underline{y})$ of $\text{Rel-Cons-SNSP}(\underline{\mathcal{C}})$ of the same objective function value.

First, we set $\underline{y} = \bar{y}$ and $\underline{\gamma} = \bar{\gamma}$. Next, $\forall (i, j) \in \mathcal{A}$, consider all sets $C^{Cons} \in \mathcal{C}_{ij}$ such that $\bar{w}_{C^{Cons}} = 1$. If $\exists \underline{C}^{Rel} \in \underline{\mathcal{C}}_{ij}$ such that $\underline{C}^{Rel} = C^{Cons}$, then set $\underline{w}_{\underline{C}^{Rel}} = 1$. If such a set \underline{C}^{Rel} does not exist then let $\underline{C}^{Rel-min}$ denote a minimal set in $\underline{\mathcal{C}}_{ij}$ such that $C^{Cons} \subseteq \underline{C}^{Rel-min}$. Such a set $\underline{C}^{Rel-min}$ must exist as either C^{Cons} is maximal, contradicting a premise of the theorem, or is contained in a maximal consolidation that is a member of the set \mathcal{C}_{ij}^{max} . If there is more than one such minimal set, one is chosen at random. We then set $\underline{w}_{\underline{C}^{Rel-min}} = 1$. Next, $\forall \underline{C}' \in \underline{\mathcal{C}}_{ij}$ such that we have not yet set $\underline{w}_{\underline{C}'} = 1$ we set $\underline{w}_{\underline{C}'} = 0$. Observe that by construction, each consolidation C^{Cons} chosen in the solution to $\text{Cons-SNSP}(\mathcal{C})$ is mapped to a single consolidation $\underline{C}^{Rel} \in \underline{\mathcal{C}}_{ij}$ such that $\underline{w}_{\underline{C}^{Rel}} = 1$. However, the mapping is not one-to-one. There may be multiple consolidations C^{Cons} chosen in the solution to $\text{Cons-SNSP}(\mathcal{C})$ that map to the same $\underline{C}^{Rel} \in \underline{\mathcal{C}}_{ij}$ and such that $\underline{w}_{\underline{C}^{Rel}} = 1$.

We next show that $(\underline{\gamma}, \underline{w}, \underline{y})$ satisfy constraints (18) - (25). Regarding constraints (18) we first recall that the values \bar{w} satisfy constraints (8), meaning that for each $k \in \mathcal{K}, (i, j) \in \mathcal{A}^k, \exists C \in \mathcal{C}_{ij}$ such that $k \in C$ and $\bar{w}_C = 1$. Next, we note that by construction, for each $C \in \mathcal{C}_{ij}$ such that $\bar{w}_C = 1$ there is at least one $\underline{C}^{Rel} \in \underline{\mathcal{C}}_{ij}, C \subseteq \underline{C}^{Rel}$, such that $\underline{w}_{\underline{C}^{Rel}} = 1$. Thus, we can conclude that for each $k \in \mathcal{K}, (i, j) \in \mathcal{A}^k, \exists \underline{C}^{Rel}$ such that $k \in \underline{C}^{Rel}$ and $\underline{w}_{\underline{C}^{Rel}} = 1$. Thus, constraints (18) are satisfied.

Next, consider constraints (19). We observe that as the values (\bar{w}, \bar{y}) satisfy constraints (9) we have that $\sum_{C \in \mathcal{C}_{ij}} s_C \bar{w}_C \leq \bar{y}_{ij}$. For a given C^{Cons} such that $\bar{w}_{C^{Cons}} = 1$, we consider the corresponding set $\underline{C}^{Rel} \in \mathcal{C}_{ij}$ such that $\underline{w}_{\underline{C}^{Rel}} = 1$ and the value $\underline{s}_{\underline{C}^{Rel}}$. Recalling equation (17) that determines the values $\underline{s}_{\underline{C}}$ we consider two cases. The first case is when $C^{Cons} \subset \underline{C}^{Rel}$. In this case, we must have that $C^{Cons} \notin \mathcal{C}_{ij}$ and $\underline{s}_{\underline{C}^{Rel}} = \min_{C' \subseteq \underline{C}^{Rel}: C' \notin \mathcal{C}_{ij}} s_{C'} \leq s_{C^{Cons}}$. The second case is when $C^{Cons} = \underline{C}^{Rel}$. In this case, we have $\underline{s}_{\underline{C}^{Rel}} \leq s_{C^{Cons}}$. As a result, we have $\underline{y}_{ij} = \bar{y}_{ij} \geq \sum_{C \in \mathcal{C}_{ij}} s_C \bar{w}_C \geq \sum_{\underline{C} \in \mathcal{C}_{ij}} \underline{s}_{\underline{C}} \underline{w}_{\underline{C}}$ and constraints (19) are satisfied.

Next, we consider constraints (20). We consider the set $\underline{C}^{Rel} \in \mathcal{C}_{ij}$ such that $\underline{w}_{\underline{C}^{Rel}} = 1$ and pairs of commodities $k, k' \in \underline{C}^{Rel}$ such that constraints (20) are defined. We have $\underline{w}_{\underline{C}^{Rel}} = 1$ because $\exists C^{Cons} \in \mathcal{C}_{ij}, C^{Cons} \subseteq \underline{C}^{Rel}$, such that $\bar{w}_{C^{Cons}} = 1$. Because constraints (20) are defined and \underline{C}^{Rel} is chosen to be the minimal set that contains C^{Cons} we must have $k, k' \in C^{Cons}$. Thus, we must have $\bar{\gamma}_{ij}^k = \bar{\gamma}_{ij}^{k'}$ as $\bar{\gamma}$ satisfy constraints (10). Given that $\underline{\gamma} = \bar{\gamma}$ we have that $\underline{\gamma}$ satisfy constraints (20). Lastly, as $\bar{\gamma}$ satisfy constraints (11) and (12) and $\underline{\gamma} = \bar{\gamma}$, $\underline{\gamma}$ satisfy constraints (21) and (22). Clearly, $(\underline{\gamma}, \underline{w}, \underline{y})$ satisfy the domain constraints (23), (24), and (25).

Thus, $(\underline{\gamma}, \underline{w}, \underline{y})$ is a feasible solution to *Rel-Cons-SNSP*(\mathcal{C}) and as $\underline{y} = \bar{y}$ the objective values of $(\underline{\gamma}, \underline{w}, \underline{y})$ and $(\bar{\gamma}, \bar{w}, \bar{y})$ in the respective optimization problems are the same. As every feasible solution to *Cons-SNSP*(\mathcal{C}) can be mapped to a feasible solution of *Rel-Cons-SNSP*(\mathcal{C}) of the same cost, we have that *Rel-Cons-SNSP*(\mathcal{C}) is a relaxation of *Cons-SNSP*(\mathcal{C}). \square \square

In summary, there are three characteristics of the formulation that render it a relaxation, assuming a suitable set \mathcal{C} as indicated in Theorem 3. The first is that constraints (18) allow a commodity to appear in more than one consolidation on an arc in its path. The second is that the computation of $\underline{s}_{\underline{C}}$ may underestimate the number of vehicles required to transport a consolidation. The third is that potentially defining constraints (20) over subsets of pairs of commodities in a consolidation allows for two commodities to appear in a chosen consolidation on an arc but dispatch at different times.

4.2 Refining the relaxation with additional consolidations

As seen in the previous section, there are three characteristics of *Rel-Cons-SNSP*(\mathcal{C}) that make it a relaxation of *Cons-SNSP*(\mathcal{C}), even though it may not consider the entire set of consolidations \mathcal{C} . Relatedly, there are three corresponding attributes of a solution $(\underline{\gamma}, \underline{w}, \underline{y})$ to *Rel-Cons-SNSP*(\mathcal{C}) that can render it infeasible for *Cons-SNSP*(\mathcal{C}). We next discuss for each attribute the method that is executed to add consolidations to \mathcal{C} to render such a solution infeasible when *Rel-Cons-SNSP*(\mathcal{C}) is next solved. We note that two of the methods rely on first finding the power set, $\mathcal{P}(C_{ij})$, of commodities in a consolidation, C_{ij} , for an arc.

Method 1: Insufficient capacity installed on an arc: Given equation (17) we know that $\underline{s}_{\underline{C}} \leq s_{\underline{C}}$. As such, while we must have $\underline{y}_{ij} \geq \sum_{\underline{C} \in \mathcal{C}_{ij}} \underline{s}_{\underline{C}} \underline{w}_{\underline{C}}$, we may also have $\sum_{\underline{C} \in \mathcal{C}_{ij}} s_{\underline{C}} \underline{w}_{\underline{C}} > \underline{y}_{ij}$. In other words, because *Rel-Cons-SNSP*(\mathcal{C}) may under-estimate the number of vehicles needed to transport a consolidation on an arc, its solution may prescribe too few vehicles for the consolidations chosen for that arc.

In this case, for each $\underline{C} \in \mathcal{C}_{ij}$ such that $\underline{s}_{\underline{C}} < s_{\underline{C}}$ and $\underline{w}_{\underline{C}} = 1$ we add sufficient missing consolidations in $\mathcal{P}(C_{ij})$ to \mathcal{C}_{ij} to ensure that after doing so $\underline{s}_{\underline{C}}$ increases by at least one vehicle. More precisely, if $\underline{s}_{\underline{C}}^t$ is the value at iteration t then the method identifies all $C \subset \underline{C}$ such that $C \notin \mathcal{C}_{ij}$ and $s_C \leq \underline{s}_{\underline{C}}^t + 1$ and adds those consolidations to \mathcal{C}_{ij} . After doing so, $\underline{s}_{\underline{C}}^{t+1} \geq \underline{s}_{\underline{C}}^t + 1$. Alternately, we could add all missing consolidations in $\mathcal{P}(\underline{C})$, leaving $\underline{s}_{\underline{C}}^{t+1} = s_{\underline{C}}$ for each such \underline{C} . In either case, when *Rel-Cons-SNSP*(\mathcal{C}) is next solved, the values $(\underline{w}, \underline{y})$ in the current solution no longer satisfy $\underline{y}_{ij} \geq \sum_{\underline{C} \in \mathcal{C}_{ij}} \underline{s}_{\underline{C}} \underline{w}_{\underline{C}}$.

Returning to our example above, we have that $\underline{y}_{ij} = 2$, but $s_{\{3,4\}} = 1$ and $s_{\{1,2,3\}} = 2$. As such, we have $s_{\{1,2,3\}} \underline{w}_{\{1,2,3\}} + s_{\{3,4\}} \underline{w}_{\{3,4\}} = 2 + 1 = 3 > \underline{y}_{ij}$. As it is consolidation $\{1,2,3\}$ such that $\underline{s}_{\{1,2,3\}} < s_{\{1,2,3\}}$, the method adds consolidations $\{1,2\}$, $\{1,3\}$, and $\{2,3\}$ to \mathcal{C}_{ij} to increase $\underline{s}_{\{1,2,3\}}$ to 2.

Method 2: Commodity appears in multiple consolidations on an arc: Consider a commodity k and arc $(i, j) \in \mathcal{A}^k$ such that $\exists \underline{C}^g, \underline{C}^h \in \mathcal{C}_{ij}$, $\underline{w}_{\underline{C}^g} = \underline{w}_{\underline{C}^h} = 1$, and $k \in \underline{C}^g \cap \underline{C}^h$. In other words, the commodity k appears in multiple chosen consolidations for the same arc. Such values would cause \underline{w} to violate constraints (8) of *Cons-SNSP*(\mathcal{C}). The method for handling such a solution is based on the following proposition which identifies how such a solution can be transformed to another solution that

is feasible to the relaxation, of equal or lesser objective function value, and with fewer violations of constraints (8).

Proposition 2. Suppose $\underline{C} \in \mathcal{C}_{ij}$ such that $\forall \underline{C}^d \subseteq \underline{C}, \underline{C}^d \in \mathcal{C}_{ij}$. Any solution $(\gamma, \underline{w}, \underline{y})$ to $Rel-Cons-SNSP(\mathcal{C})$ such that $\underline{w}_{\underline{C}^g} = \underline{w}_{\underline{C}^h} = 1, \underline{C}^g, \underline{C}^h \subseteq \mathcal{C}_{ij}$ and $\underline{C}^g \cap \underline{C}^h \neq \emptyset$ can be transformed to another feasible solution $(\gamma', \underline{w}', \underline{y}')$ of equal or lesser objective function value such that for all $\underline{C}^m, \underline{C}^n \subseteq \mathcal{C}_{ij}$ wherein $\underline{w}'_{\underline{C}^m} = \underline{w}'_{\underline{C}^n} = 1$, we have $\underline{C}^m \cap \underline{C}^n = \emptyset$.

Proof. Let $\underline{C}^{int} = \underline{C}^g \cap \underline{C}^h$ denote the consolidation containing the commodities that participate in both chosen consolidations. Because we have $\forall \underline{C}^d \subseteq \underline{C}, \underline{C}^d \in \mathcal{C}_{ij}$, we have $\underline{s}_{\underline{C}^g} = s_{\underline{C}^g}$ and $\underline{s}_{\underline{C}^h} = s_{\underline{C}^h}$. Thus, we have $\underline{y}_{ij} \geq s_{\underline{C}^g} + s_{\underline{C}^h} = \lceil \frac{\sum_{k \in \underline{C}^g \setminus \underline{C}^{int}} q_k + \sum_{k \in \underline{C}^{int}} q_k}{u_{ij}} \rceil + \lceil \frac{\sum_{k \in \underline{C}^h \setminus \underline{C}^{int}} q_k + \sum_{k \in \underline{C}^{int}} q_k}{u_{ij}} \rceil \geq \lceil \frac{\sum_{k \in \underline{C}^g \setminus \underline{C}^{int}} q_k + \sum_{k \in \underline{C}^{int}} q_k}{u_{ij}} \rceil + \lceil \frac{\sum_{k \in \underline{C}^h \setminus \underline{C}^{int}} q_k}{u_{ij}} \rceil$.

Because we have $\forall \underline{C}^d \subseteq \underline{C}, \underline{C}^d \in \mathcal{C}_{ij}$ we have that $\underline{C}^h \setminus \underline{C}^{int} \in \mathcal{C}_{ij}$. Thus, the variable $w_{\underline{C}^h \setminus \underline{C}^{int}}$ is defined for $Rel-Cons-Sched(\mathcal{C})$ and setting $\underline{w}_{\underline{C}^h} = 0, \underline{w}_{\underline{C}^h \setminus \underline{C}^{int}} = 1$ and setting remaining variables to their values in $(\gamma, \underline{w}, \underline{y})$ yields a feasible solution of equal or lesser objective function value and one fewer violation of constraints (8). \square

Thus, when we encounter such a solution to the $Rel-Cons-SNSP(\mathcal{C})$ we construct the set $\underline{C}^{u-gh} = \underline{C}^g \cup \underline{C}^h$. If we have $\underline{C}^m \in \mathcal{C}_{ij}, \forall \underline{C}^m \subseteq \underline{C}^{u-gh}$ we construct the alternative feasible solution as in the proof of Proposition 2. Otherwise, we create all missing consolidations in $\mathcal{P}(\underline{C}^{u-gh})$ and add them to \mathcal{C}_{ij} .

Returning to our example, we have $\underline{C}^g = \{1, 2, 3\}$ and $\underline{C}^h = \{3, 4\}$ and thus $\underline{C}^{u-gh} = \{1, 2, 3, 4\}$. The method would then add the consolidations $\{1, 2\}, \{1, 3\}$, and $\{2, 3\}$ to \mathcal{C}_{ij} .

Method 3: Pair of commodities in chosen consolidation but dispatch at different times: Consider an arc $(i, j) \in \mathcal{A}$ and consolidation $\underline{C} \in \mathcal{C}_{ij}$ such that $\underline{w}_{\underline{C}} = 1$. Next, suppose there exist commodities $k, k' \in \underline{C}$ such that $\gamma_{ij}^k \neq \gamma_{ij}^{k'}$, causing $\underline{w}_{\underline{C}_{ij}}, \gamma_{ij}^k$, and $\gamma_{ij}^{k'}$ to violate constraints (10) of $Cons-SNSP(\mathcal{C})$. Given the definition of constraints (20) in $Rel-Cons-SNSP(\mathcal{C})$, we must have either $\underline{C} \setminus k \notin \mathcal{C}_{ij}$, or $\underline{C} \setminus k' \notin \mathcal{C}_{ij}$, or both. Such a solution can be rendered infeasible for $Rel-Cons-SNSP(\mathcal{C})$ by adding to \mathcal{C}_{ij} whichever of the consolidations $\underline{C} \setminus k, \underline{C} \setminus k'$ are missing.

Returning to our example, we see that $\underline{w}_{\{1,2,3\}} = 1$ but $\gamma_{ij}^1 = \gamma_{ij}^2 = 1$ and $\gamma_{ij}^3 = 3$. This is feasible for $Rel-Cons-SNSP(\mathcal{C})$ because consolidations $\{1, 2, 3\} \setminus \{3\} = \{1, 2\}$ and $\{1, 2, 3\} \setminus \{2\} = \{1, 3\}$ are not contained in \mathcal{C}_{ij} and thus constraints (20) are not present in $Rel-Cons-SNSP(\mathcal{C})$. Thus, the method would add consolidations $\{1, 2\}$ and $\{1, 3\}$ to \mathcal{C}_{ij} .

4.3 Algorithm description and proof of correctness

In this section, we present a formal description of the algorithm we propose for solving $Cons-SNSP(\mathcal{C})$, which we refer to as $IP-ColGen$. We note that we include in the description some steps that are not necessary for its correctness, but speed up its convergence. For example, at each iteration, the algorithm solves $Cons-SNSP(\mathcal{C})$ to heuristically generate a primal solution. In addition, by ensuring that \mathcal{C}_{ij} contains both maximal consolidations and singletons, the $Cons-SNSP(\mathcal{C})$ solved at an iteration is guaranteed to be feasible. We note that through a tuning exercise we determined that performing the refinement methods in the order Method 3, Method 1, and then Method 2 lead to the best performance of the algorithm.

In the interest of brevity, we do not provide a detailed description of how Step 6 of $IP-ColGen$ is performed. However, we note that identifying aspects of a solution $(\gamma, \underline{w}, \underline{y})$ to the relaxation that render it infeasible for $Cons-SNSP(\mathcal{C})$ can be done via simple inspection. For example, checking to see whether constraints (8) are satisfied can be done by looking at all consolidations $\underline{C}^g, \underline{C}^h \in \mathcal{C}_{ij}$ such that $\underline{w}_{\underline{C}^g} = \underline{w}_{\underline{C}^h} = 1$ and checking that their intersection is empty. Checking whether constraints (9) are satisfied can be done by computing $\sum_{\underline{C} \in \mathcal{C}_{ij}} s_{\underline{C}} \underline{w}_{\underline{C}}$ and observing whether it exceeds the value \underline{y}_{ij} . Finally, checking whether constraints (10) are satisfied can be done by looking at all consolidations $\underline{C} \in \mathcal{C}_{ij}$ such that $\underline{w}_{\underline{C}} = 1$ and all commodities $k, k' \in \underline{C}$ and checking whether $\gamma_{ij}^k = \gamma_{ij}^{k'}$.

We next prove the correctness of $IP-ColGen$.

Proposition 3. $IP-ColGen$ is guaranteed to terminate with an optimal solution to $Cons-SNSP(\mathcal{C})$ in a finite number of iterations.

Algorithm 1 IP-COLGEN

Input: Time limit τ , Optimality tolerance ϵ

Input: Flat network $\mathcal{D} = (\mathcal{N}, \mathcal{A})$, commodity set \mathcal{K}

```
1: for all  $(i, j) \in \mathcal{A}$  do
2:   Set  $\underline{\mathcal{C}}_{ij} = \underline{\mathcal{C}}_{ij}^{max} \cup_{k \in \mathcal{K}_{ij}} \{k\}$  {Initialize consolidation set for each arc with maximal consolidations and singletons}
3: end for
4: while not solved and elapsed time  $< \tau$  do
5:   Solve  $Rel-Cons-SNSP(\underline{\mathcal{C}})$  for dual bound  $lb_{Cons}$ 
6:   Determine whether solution to  $Rel-Cons-SNDP(\underline{\mathcal{C}})$  is feasible for  $Cons-SNSP(\mathcal{C})$ 
7:   if feasible then
8:     Stop. Solution is optimal.
9:   else
10:    Add consolidations to  $\underline{\mathcal{C}}$  via Method 3 to rectify consolidated commodities that dispatch at different times
11:    Add consolidations to  $\underline{\mathcal{C}}$  via Method 1 to rectify insufficient capacity
12:    Add consolidations to  $\underline{\mathcal{C}}$  via Method 2 to rectify commodities that appear in multiple consolidations
13:   end if
14:   Solve  $Cons-SNSP(\underline{\mathcal{C}})$  for primal bound  $ub_{Cons}$ 
15:   if  $(ub_{Cons} - lb_{Cons})/ub_{Cons} < \epsilon$  then
16:     Stop. Current best primal solution is  $\epsilon$ -optimal.
17:   end if
18: end while
```

Proof. As $Rel-Cons-SNSP(\underline{\mathcal{C}})$ is a relaxation of $Cons-SNSP(\mathcal{C})$ any optimal solution to $Rel-Cons-SNSP(\underline{\mathcal{C}})$ that is feasible for $Cons-SNSP(\mathcal{C})$ is also optimal for $Cons-SNSP(\mathcal{C})$. However, when an optimal solution to $Rel-Cons-SNSP(\underline{\mathcal{C}})$ is not feasible for $Cons-SNSP(\mathcal{C})$, at least one of Steps 10, 11, or 12 of $IP-ColGen$ will add a consolidation to $\underline{\mathcal{C}}$. As consolidations are never removed from $\underline{\mathcal{C}}$ and \mathcal{C} is finite, $IP-ColGen$ must ultimately produce a set $\underline{\mathcal{C}}$ that is equal to \mathcal{C} .

When $\underline{\mathcal{C}} = \mathcal{C}$, we have $\underline{s}_{\underline{\mathcal{C}}} = s_{\mathcal{C}} \quad \forall (i, j) \in \mathcal{A}, \underline{\mathcal{C}} \in \underline{\mathcal{C}}$. We also have constraints (20) defined for all $k, k' \in \underline{\mathcal{C}}, (i, j) \in \mathcal{A}, \underline{\mathcal{C}} \in \underline{\mathcal{C}}$. Finally, as noted in Proposition (2), any solution to $Rel-Cons-SNSP(\underline{\mathcal{C}})$ such that a commodity participates in multiple consolidations on a given arc can be transformed to another feasible solution of equal or lesser objective function value where this does not occur. As such, any optimal solution to $Rel-Cons-SNSP(\underline{\mathcal{C}})$ is either feasible and optimal for $Cons-SNSP(\mathcal{C})$, or can be converted to a feasible and optimal solution for $Cons-SNSP(\mathcal{C})$. \square \square

4.4 Other enhancements

In this section, we identify other enhancements to $IP-ColGen$ that computational experiments indicated improved its performance. These enhancements involve adding additional consolidations to $\underline{\mathcal{C}}$ at an iteration above and beyond what is necessary for its correctness.

Adding consolidations based on commodities that dispatch at the same time: We recall that in a solution to $Rel-Cons-SNSP(\underline{\mathcal{C}})$ we may have $\underline{w}_{\underline{\mathcal{C}}} = 1$ for consolidation $\underline{\mathcal{C}}$ but not all commodities $k \in \underline{\mathcal{C}}$ dispatch at the same time. In this case, we partition $\underline{\mathcal{C}}$ into sets of commodities that dispatch at the same time in the solution and add a consolidation to $\underline{\mathcal{C}}_{ij}$ for each set of commodities that is not in $\underline{\mathcal{C}}_{ij}$. More precisely, we add sets $\underline{\mathcal{C}}^g, g = 1, \dots, n_{ij}$ such that $\underline{\mathcal{C}} = \bigcup_{g=1}^{n_{ij}} \underline{\mathcal{C}}^g, \underline{\mathcal{C}}^g \cap \underline{\mathcal{C}}^h = \emptyset, g = 1, \dots, n_{ij} - 1, h = g + 1, \dots, n_{ij}$ and $\underline{\gamma}_{ij}^k = \underline{\gamma}_{ij}^{k''} \quad \forall k, k' \in \underline{\mathcal{C}}_{ij}^g$. In this case, we add to $\underline{\mathcal{C}}_{ij}$ the sets $\underline{\mathcal{C}}^g$ that it does not currently contain. Returning to our example, we have $\underline{w}_{\{1,2,3\}} = 1$ but $\underline{\gamma}_{ij}^1 = \underline{\gamma}_{ij}^2 = 1$ and $\underline{\gamma}_{ij}^3 = 3$. As commodities 1 and 2 dispatch together in this solution and $\{1, 2\} \notin \underline{\mathcal{C}}_{ij}$, the method would add consolidation $\{1, 2\}$ to $\underline{\mathcal{C}}_{ij}$.

Adding missing subset consolidations: Recalling the computation of $\underline{s}_{\underline{\mathcal{C}}}$ in equation (17), we see that when adding consolidation $\underline{\mathcal{C}}$ to $\underline{\mathcal{C}}_{ij}$ we will have $\underline{s}_{\underline{\mathcal{C}}} < s_{\underline{\mathcal{C}}}$ if there exists a subset consolidation, $\underline{\mathcal{C}}' \subseteq \underline{\mathcal{C}}$, such that $\underline{\mathcal{C}}' \notin \underline{\mathcal{C}}_{ij}$ and $s_{\underline{\mathcal{C}}} < s_{\underline{\mathcal{C}'}}$. Thus, we add to $\underline{\mathcal{C}}_{ij}$ all consolidations $\underline{\mathcal{C}}' \subseteq \underline{\mathcal{C}}$ such that $|\underline{\mathcal{C}}'| \leq \kappa$ and $\underline{\mathcal{C}}' \notin \underline{\mathcal{C}}_{ij}$. Here, κ is an algorithm parameter.

5 Computational analysis

In this computational study we compare four methods for solving the SNSP. The first two methods are static in that they involve instantiating a mixed integer programming formulation of the SNSP that is then solved. The first, labeled *Cons-SNSP*(\mathcal{C}), consists of enumerating the set \mathcal{C} and solving the resulting consolidation-based formulation, *Cons-SNSP*(\mathcal{C}). The second, labeled “Reduced T-E Network + Agg-Arc-Capacity” consists of generating a reduced time-expanded network $\mathcal{D}_{\mathcal{T}}^R$ as described in Proposition 1 and then solving the resulting *TEN-SNSP*($\mathcal{D}_{\mathcal{T}}^R$) formulation, albeit strengthened with the Agg-Arc-Capacity inequalities (6). The second two methods are dynamic, in that they involve iteratively solving integer programs that are modified at each iteration in a manner that guarantees their solution will ultimately produce a provably optimal solution. More precisely, the third method is to solve the *Cons-SNSP*(\mathcal{C}) with the *IP – ColGen* approach described in Section 4. The fourth, labeled “DDD-SNSP,” is to solve the *TEN-SNSP*($\mathcal{D}_{\mathcal{T}}$) with the enhanced dynamic discretization discovery algorithm of Hewitt (2019). We first describe the experimental setting underlying the computational analysis and then the results of those experiments.

5.1 Computational setting

To perform our computational analysis, we generated a set of instances based on a portion of the network of a United States-based LTL carrier. Specifically, we considered a portion of the network that consists of 25 terminals (e.g. $|\mathcal{N}| = 25$) and 530 physical moves between terminals (e.g. $|\mathcal{A}| = 530$). Cost, capacity, and travel time data were provided by the carrier. In addition, the carrier provided a *load plan* that prescribed paths through the terminal network for pairs of terminals $(o, d) \in \mathcal{N}$ based on their customer base at the time. We randomly generated instances that vary in the number of commodities (100, 150, 200, 250, 300, 350, 400, 450, and 500) and the width $l_k - e_k$ of the delivery window of each commodity. Specifically, we considered four widths, with the first corresponding to the delivery window quoted by the carrier. The remaining three consist of the length of that path (in time) plus three, six, or nine hours. For each of the nine numbers of commodities and four delivery window widths, we generated five instances. When doing so, the size of each commodity was randomly drawn from distributions fit to carrier historical data of shipment sizes. In total, our instance set consists of 225 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. The DDD approach was implemented in C++, which solved optimization models with CPLEX 12.10 (Studio-CPLEX 2013). For the other approaches, the formulations and code to instantiate them were implemented in Python 3.7 (VanRossum and Drake 2010). For those approaches, optimization models were solved with CPLEX 12.10 (Studio-CPLEX 2013). All code was run for two hours, or, 7,200 seconds. All optimization models were solved to a 1% tolerance. MIPs solved in the course of executing *IP – ColGen* were solved with the same limits. When executing *IP – ColGen* a limit of 5,000 was placed on the total number of consolidations added to \mathcal{C} at an iteration. The parameter κ for that algorithm was given the value 7. The values of 5,000 and 7 were determined through a parameter tuning exercise. All times reported are in seconds.

5.2 Computational results

We first compare the four methods on two dimensions: (1) the percentage of instances the method could solve within the 7,200 second time limit, and, (2) the average time the method needed to terminate. We report these percentages and times to termination, averaged over all instances, in Table 6.

Table 6: % solved, time to termination, averaged over all instances

Type	Method	% Solved	Time (sec.)
Static	<i>Cons-SNSP</i> (\mathcal{C})	77.78%	1,770.94
	Reduced T-E Network + Agg-Arc-Capacity	66.67%	2,531.47
Dynamic	<i>IP – ColGen</i>	100.00%	28.40
	DDD-SNSP	100.00%	97.63

We see that of the two static methods, solving the consolidation-based formulation enabled the largest percentage of instances to be solved and required less time to do so. However, neither static method is

able to solve every instance. We also see that both dynamic methods are able to solve every instance. In addition, of the two dynamic methods, $IP - ColGen$ does so in less than a third of the time required by DDD-SNSP.

We next report in Table 7 the average times to completion for each method and each number of commodities in an instance. We see that for every number of commodities, $IP - ColGen$ requires the least time to finish. For instances with 250 commodities or fewer, $Cons-SNSP(\mathcal{C})$ is the second fastest method. However, for instances with more than 250 commodities, DDD-SNSP is the second fastest method.

Table 7: Average time to termination (seconds)

# Commodities	Static		Dynamic	
	$Cons-SNSP(\mathcal{C})$	Reduced T-E Network + Agg-Arc-Capacity	$IP - ColGen$	DDD-SNSP
100	0.52	8.09	0.52	26.05
150	1.41	34.26	1.10	43.90
200	3.00	59.16	1.58	51.25
250	5.43	1,856.19	2.79	73.15
300	1,807.23	521.28	6.52	99.90
350	1,857.88	2,110.91	8.16	112.65
400	606.80	5,516.23	28.56	133.75
450	5,462.01	5,477.16	75.16	161.35
500	6,194.21	7,200.00	131.18	176.70
Average	1,770.94	2,531.47	28.40	97.63

We also see that the time required by $Cons-SNSP(\mathcal{C})$ rises dramatically as the number of commodities in an instance increases. Not surprisingly, this can be attributed to the need to enumerate all possible consolidations of commodities in order to formulate $Cons-SNSP(\mathcal{C})$. We study the effort associated with this enumeration in Table 8. This table reports, by number of commodities in an instance, the percentage of instances wherein all consolidations could be enumerated within the two hour time limit. It also reports the average time needed for the enumeration when it could be completed within the two hour limit. We see a general rise in the average time needed by the enumeration as the number of commodities in an instance increases. For the 300 and 350 commodity instances, we observe that it is impossible to enumerate all consolidations for the instances with the widest delivery windows. For the 450 and 500 commodity instances, the feasible consolidations could be enumerated only for the narrowest delivery window setting.

Table 8: Effort to enumerate consolidations

$ \mathcal{K} $	100	150	200	250	300	350	400	450	500	Average
% instances	100.00%	100.00%	100.00%	100.00%	75.00%	75.00%	100.00%	25.00%	25.00%	77.78%
Time (sec.)	0.09	0.36	1.75	3.56	6.58	67.76	583.30	232.15	3,143.07	212.66

However, Table 7 indicates that formulating and solving the $Cons-SNSP(\mathcal{C})$ can be computationally effective when the number of commodities in an instance is not too large. To understand why, we formulated and solved the linear relaxations associated with the two static methods. We compare the objective function value, z_{LP}^{Meth} , of the linear relaxation associated with method $Meth$ (e.g. $Cons - SNSP(\mathcal{C})$) with the objective function value, $z_{Cons}(\mathcal{C})$, of the optimal solution to the integer program $Cons-SNSP(\mathcal{C})$. We report in Table 9 averages of the gap $(z_{Cons}(\mathcal{C}) - z_{LP}^{Meth})/z_{Cons}(\mathcal{C})$ and the time required to formulate and solve linear relaxations. We note we only consider instances wherein the set \mathcal{C} could be enumerated and the resulting linear programming relaxation of $Cons-SNSP(\mathcal{C})$ solved in the two hours time limit. We see that solving the $Cons-SNSP(\mathcal{C})$ yields a much stronger lower bound than the linear relaxation of $TEN-SNSP(\mathcal{D}_T^R)$. However, it also takes much more time to solve. Note that the time recorded for solving the linear programming relaxation of $Cons-SNSP(\mathcal{C})$ includes the time needed to enumerate the consolidations needed by the formulation.

Table 9: Strength of LP bound and time to solve LP, averaged over all instances

	Reduced T-E Network + Agg-Arc-Capacity	$Cons-SNSP(\mathcal{C})$
Gap with best IP sol'n	5.23%	1.64%
Time (sec.) to solve LP	186.10	1,047.20

We next turn our attention to the two dynamic methods. Another parameter defining an instance is the number of hours added to the delivery window offered by the carrier. To understand the impact of this parameter value on the performance of the two dynamic methods we report in Table 10 the average time to termination, averaged over instances with the same number of additional hours.

We see that the behavior of DDD-SNSP and $IP - ColGen$ differs with respect to this parameter. Specifically, for $IP - ColGen$ the time it requires to solve an instance increases in the number of hours added to a delivery window. However, for DDD-SNSP, the time (mostly) decreases. That the time required by $IP - ColGen$ increases is likely due to the fact that the wider the delivery windows the greater the number of consolidations that the algorithm may need to consider. On the other hand, DDD-SNSP requires fewer iterations when the delivery windows are wide.

Table 10: Average time to termination, by number of additional hours in delivery window

Hours added to delivery window	$IP - ColGen$	DDD-SNSP
0	11.59	125.07
3	24.12	90.18
6	22.97	86.6
9	54.91	88.69

Next, we analyze in Table 11 the performance of $IP - ColGen$ in more detail. In column “# Iterations” of Table 11 we report the average number of iterations $IP - ColGen$ executed before termination. We also report in Table 11 the quality of the first lower bound and primal solution produced by $IP - ColGen$. Specifically, if lb_{first} is the lower bound derived from solving $Rel-Cons-SNSP(\mathcal{C})$ in the first iteration and ub_{last} is the upper bound reported at its termination then we measure the quality of the initial lower bound with the gap $(ub_{last} - lb_{first})/ub_{last}$. With lb_{last} and ub_{first} defined similarly, we measure the quality of the initial upper bound produced by $IP - ColGen$ with the gap $(ub_{first} - lb_{last})/ub_{first}$.

Table 11: Performance of $IP - ColGen$ in detail

# Commodities	Iterations	Quality init. LB	Quality init. UB
100	1	0.22%	0.22%
150	1	0.24%	0.24%
200	1	0.09%	0.09%
250	1	0.41%	0.41%
300	1.25	0.30%	0.60%
350	1.25	0.69%	0.84%
400	2	0.98%	1.06%
450	1.75	1.30%	1.26%
500	3.25	1.39%	1.13%
Average	1.5	0.63%	0.65%

We first observe that $IP - ColGen$ is able to produce a provably optimal solution in a small number of iterations. This is likely because the quality of the initial lower and upper bounds $IP - ColGen$ produces are quite high, with the lower bound of slightly higher quality than the upper bound. On average, after its first iteration $IP - ColGen$ has found a primal solution and dual bound that together yield an optimality gap of 0.85%.

6 Conclusion and future work

In this paper, we present two mixed integer programming (MIP) formulations of a scheduling problem that can be solved to plan the operations of freight transportation carriers. More specifically, carriers that transport shipments that are small relative to vehicle capacity. Given these small sizes, the carriers seek to route and schedule shipments through a terminal network such that they may consolidate, i.e. be transported by the same vehicle. The literature generally views determining routes and schedules as a tactical planning problem, with most papers on this topic discussing some variant of the *Scheduled Service Network Design Problem* (SSNDP). We consider a more operational problem wherein shipment paths have already been determined and thus it is the shipment and vehicle schedules that need to be optimized. We call this problem the *Service Network Scheduling Problem* (SNSP).

One of the MIP formulations is based on the idea of a time-expanded network. The other is based on enumerating all sets of shipments that can dispatch at the same time when traveling from one terminal to the next (i.e. “consolidations.”). With a computational study we show that when enumerating all such sets is not too time-consuming, solving the consolidation-based formulation was much more effective. This is in part because its linear programming relaxation provides a much stronger lower bound on the objective function value of the optimal solution to the integer program. However, for instances with larger sets of shipments, the enumerative nature of the consolidation-based formulation renders it computationally ineffective.

We also propose an algorithm for solving the consolidation-based formulation without generating all sets of consolidations *a priori*. The algorithm instead generates these sets dynamically and follows an algorithmic strategy that is similar in spirit to Branch-and-Price, but solves optimization problems at iterations that are integer programs instead of linear programs. We prove the correctness of this algorithm and with an extensive computational study illustrate its superior performance to the previously presented methods as well as a state-of-the-art benchmark from the literature, *Dynamic Discretization Discovery*.

Regarding future work, a Branch-and-Price-based algorithm for the problem is likely worth developing, potentially in the context of a hybrid method that alternates between Branch-and-Price and the method we propose in this paper. Solution methods can also be developed for other planning problems encountered by consolidation carriers that rely on solving the SNSP as a subroutine. Heuristics for the SSNDP could be developed that evaluate a neighboring solution by first routing commodity flows in the terminal network and then scheduling them via the SNSP. Given that heuristics often perform best when solution evaluation is fast, executing just the first iteration of *IP – ColGen* may be an effective strategy, particularly given the computational results that indicate the high quality of the primal solution and dual bound produced at the first iterations of *IP – ColGen*.

The consolidation-based formulation and algorithm proposed in this paper can be adapted/incorporated into exact methods for the SSNDP. For example, solving a hybrid formulation wherein some arcs in the network are modeled on a time-expanded network and others are modeled as in the *Cons-SNSP(C)* may be computationally effective. Such a formulation may be particularly appropriate for carriers that operate networks that have a hub-and-spoke-type structure, with spokes connecting to few hubs. With such a network, it is not uncommon for all potential paths for a shipment that originates at a spoke to first move to the same hub. Thus, the arc that models this first move is amenable to the consolidation-based formulation of the *Cons-SNSP(C)*. A similar observation can be made regarding shipments that are destined for a common spoke.

Acknowledgment

We would like to acknowledge the support of a 2018 Cross Regional Grant from the INFORMS Transportation Science & Logistics Society.

References

- Jardar Andersen, Teodor Gabriel Crainic, and Marielle Christiansen. Service network design with asset management: Formulations and comparative analyses. *Transportation Research Part C: Emerging Technologies*, 17(2):197–207, 2009.
- Jardar Andersen, Marielle Christiansen, Teodor Gabriel Crainic, and Roar Grønhaug. Branch and price for service network design with asset management constraints. *Transportation Science*, 45(1):33–49, 2011.
- Cynthia Barnhart, Ellis L Johnson, George L Nemhauser, Martin WP Savelsbergh, and Pamela H Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3):316–329, 1998.
- Ahmad Baubaid, Natasha Boland, and Martin Savelsbergh. The value of limited flexibility in service network designs. *Transportation Science*, 2020.
- J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.
- Natasha Boland, Mike Hewitt, Luke Marshall, and Martin Savelsbergh. The continuous-time service network design problem. *Operations Research*, 65(5):1303–1321, 2017a.
- Natasha Boland, Mike Hewitt, Luke Marshall, and Martin Savelsbergh. The prize of discretizing time: A study in service network design. *Submitted*, 2017b.
- Mervat Chouman and Teodor Gabriel Crainic. Cutting-plane matheuristic for service network design with design-balanced requirements. *Transportation Science*, to appear.

- Teodor Gabriel Crainic and Antonino Sgalambro. Service network design models for two-tier city logistics. *Optimization Letters*, 8(4):1375–1387, 2014.
- Teodor Gabriel Crainic, Mike Hewitt, Michel Toulouse, and Duc Minh Vu. Service network design with resource constraints. *Transportation Science*, 50(4):1380–1393, 2014.
- T.G. Crainic. Service network design in freight transportation. *European Journal of Operations Research*, 122(2):272–288, 2000.
- Guy Desaulniers, Jacques Desrosiers, and Marius M Solomon. *Column generation*, volume 5. Springer Science & Business Media, 2006.
- Alan Erera, Michael Hewitt, Martin Savelsbergh, and Yang Zhang. Improved load plan design through integer programming based local search. *Transportation Science*, 47(3):412–427, 2013.
- Lester Randolph Ford and Delbert Ray Fulkerson. Constructing maximal dynamic flows from static flows. *Operations research*, 6(3):419–433, 1958.
- Lester Randolph Ford and Delbert Ray Fulkerson. *Flows in networks*. Princeton University Press, 1962.
- Mike Hewitt. Enhanced dynamic discretization discovery for the continuous time load plan design problem. *Transportation Science*, 53(6):1731–1750, 2019.
- Mike Hewitt, Teodor Gabriel Crainic, Maciek Nowak, and Walter Rei. Scheduled service network design with resource acquisition and management under uncertainty. *Transportation Research Part B: Methodological*, 128:324–343, 2019.
- Ahmad Jarrah, Ellis Johnson, and Lucas Neubert. Large-scale, less-than-truckload service network design. *Operations Research*, 57(3), 2009.
- Luke Marshall, Natashaia Boland, Martin Savelsbergh, and Mike Hewitt. Interval-based dynamic discretization discovery for solving the continuous-time service network design problem. *Transportation Science*, 2020.
- George L Nemhauser and Laurence A Wolsey. *Integer and combinatorial optimization*, volume 18. Wiley New York, 1988.
- Michael Pinedo. *Scheduling*, volume 29. Springer, 2012.
- Elina Rönnberg and Torbjörn Larsson. Column generation in the integral simplex method. *European Journal of Operational Research*, 192(1):333–342, 2009.
- Elina Rönnberg and Torbjörn Larsson. All-integer column generation for set partitioning: Basic principles and extensions. *European Journal of Operational Research*, 233(3):529–538, 2014.
- Martin Savelsbergh. A branch-and-price algorithm for the generalized assignment problem. *Operations research*, 45(6):831–841, 1997.
- Yannick Oskar Scherr, Bruno Albert Neumann Saavedra, Mike Hewitt, and Dirk Christian Mattfeld. Service network design with mixed autonomous fleets. *Transportation Research Part E: Logistics and Transportation Review*, 124:40–55, 2019.
- Yannick Oskar Scherr, Mike Hewitt, Bruno Albert Neumann Saavedra, and Dirk Christian Mattfeld. Dynamic discretization discovery for the service network design problem with mixed autonomous fleets. *Transportation Research Part B: Methodological*, 141:164–195, 2020.
- John D. Schulz. Rival carriers chase former central freight customers as tight ltl market shrinks, December 2021. URL https://www.logisticsmgmt.com/article/rival_carriers_chase_former_central_freight_customers_as_tight_ltl_market_s.
- IBM ILOG CPLEX Optimization Studio-CPLEX. Users manual-version 12 release 6, 2013.
- Adil Tahir, Guy Desaulniers, and Issmail El Hallaoui. Integral column generation for the set partitioning problem. *EURO Journal on Transportation and Logistics*, 8(5):713–744, 2019.
- Adil Tahir, Frédéric Quesnel, Guy Desaulniers, Issmail El Hallaoui, and Yassine Yaakoubi. An improved integral column generation algorithm using machine learning for aircrew pairing. *Transportation Science*, 55(6):1411–1429, 2021.
- Adil Tahir, Guy Desaulniers, and Issmail El Hallaoui. Integral column generation for set partitioning problems with side constraints. *INFORMS Journal on Computing*, 2022.
- UPS. Ups annual report, December 2021. URL <https://investors.ups.com/company-profile>.
- Guido VanRossum and Fred L Drake. *The python language reference*. Python Software Foundation Amsterdam, Netherlands, 2010.
- Nicole Wieberneit. Service network design for freight transportation: a review. *OR Spectrum*, 30(1), 2008.
- Endong Zhu, Teodor Gabriel Crainic, and Michel Gendreau. Scheduled service network design for freight rail transportation. *Operations research*, 62(2):383–400, 2014.

A Proof of Theorem 1

We next provide a formal proof that solving the consolidation-based formulation yields the same optimal objective function value as solving the time expanded-network based formulation.

Theorem 4. $z_{TEN}(\mathcal{D}_{\mathcal{T}}) = z_{Cons}(\mathcal{C})$.

Proof. We begin by showing that any solution (x^{TEN}, y^{TEN}) to $TEN-SNSP(\mathcal{D}_{\mathcal{T}})$ can be transformed to a solution $(\gamma^{Cons}, w^{Cons}, y^{Cons})$ of $Cons-SNSP(\mathcal{C})$ of equal cost. First, for each $k \in \mathcal{K}$ and arc $((i, t), (j, t')) \in \mathcal{A}_{\mathcal{T}}^k$ such that $x_{ij}^{ktt'} = 1$ we set $\gamma_{ij}^{kCons} = t$. Given that $\mathcal{D}_{\mathcal{T}}$ is constructed such that when $((i, t), (j, t')) \in \mathcal{A}_{\mathcal{T}}$, $t' \geq t + \tau_{ij}$ and the presence of the flow balance constraints (2) in $TEN-SNSP(\mathcal{D}_{\mathcal{T}})$, these γ_{ij}^{kCons} values satisfy constraints (11), (12), and (13) of $Cons-SNSP(\mathcal{C})$. Regarding vehicle moves, for all $(i, j) \in \mathcal{A}$, let $y_{ij}^{Cons} = \sum_{((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}}} y_{ij}^{tt'TEN}$. Clearly, the values y_{ij}^{Cons} satisfy constraints (15).

We next consider constraints (8) of $Cons-SNSP(\mathcal{C})$. Consider a given $(i, j) \in \mathcal{A}$ and times t, t' such that $((i, t), (j, t')) \in \mathcal{A}_{\mathcal{T}}$ and $\exists k \in \mathcal{K}_{ij}$ such that $x_{ij}^{ktt'TEN} = 1$. For such a t , consider the maximal set of commodities $\mathcal{K}_{ij}^t = \{k' \in \mathcal{K}_{ij} : x_{ij}^{k'tt'TEN} = 1\}$ that dispatch at time t . In other words, $\forall k'' \in \mathcal{K}_{ij} \setminus \mathcal{K}_{ij}^t, x_{ij}^{k''tt'} = 0$. As all the time windows for commodities in \mathcal{K}_{ij}^t overlap in the time t , there must exist a set $C_{ij}^{qt} \in \mathcal{C}_{ij}$ that equals \mathcal{K}_{ij}^t . Let $w_{C_{ij}^{qt}}^{Cons} = 1$. Next, consider $k \in \mathcal{K}_{ij}^t$. Note that because the values $x_{ij}^{ktt'TEN}$ satisfy the flow balance constraints (2) in $TEN-SNSP(\mathcal{D}_{\mathcal{T}})$ associated with nodes of the form $(i, \cdot) \in \mathcal{N}_{\mathcal{T}}$, there is exactly one t such that $x_{ij}^{ktt'TEN} = 1$. Since \mathcal{K}_{ij}^t is maximal, there is exactly one set C_{ij}^{qt} such that $w_{C_{ij}^{qt}}^{Cons} = 1$ for time t . Thus, we must have $\sum_{C_{ij} \in \mathcal{C}_{ij}} \phi_{C_{ij}}^k w_{C_{ij}} = 1 \quad \forall k \in \mathcal{K}_{ij}$. More generally, because the values $x_{ij}^{ktt'TEN}$ satisfy the flow balance constraints (2) in $TEN-SNSP(\mathcal{D}_{\mathcal{T}})$, for a given k and arc (i, j) in its path there must be a single t such that $x_{ij}^{ktt'TEN} = 1$. Hence, these w^{Cons} values satisfy constraints (8) of $Cons-SNSP(\mathcal{C})$.

Similarly, note that by the construction of the variable values γ_{ij}^{kCons} , the values $\gamma_{ij}^{kCons}, \gamma_{ij}^{k'Cons}, w_{C_{ij}^{qt}}$ satisfy constraints (10) for all $k, k' \in C_{ij}^{qt}$. Regarding constraints (9), recall that $s_{C_{ij}^{qt}} = \lceil \sum_{k \in \mathcal{K}_{ij}^t} q_k / u_{ij} \rceil$. As (x^{TEN}, y^{TEN}) is feasible for $TEN-SNSP(\mathcal{D}_{\mathcal{T}})$ we have $y_{ij}^{tt'TEN} \geq \lceil \sum_{k' \in \mathcal{K}_{ij}^t} q_{k'} / u_{ij} \rceil$. Considering all \tilde{t} such that $\mathcal{K}_{ij}^{\tilde{t}} \neq \emptyset$ we have $y_{ij}^{Cons} = \sum_{((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}}} y_{ij}^{tt'TEN} \geq \sum_{\tilde{t} \in \mathcal{T} : \mathcal{K}_{ij}^{\tilde{t}} \neq \emptyset} y_{ij}^{\tilde{t}\tilde{t}'TEN} \geq \sum_{\tilde{t} \in \mathcal{T} : \mathcal{K}_{ij}^{\tilde{t}} \neq \emptyset} \sum_{k' \in \mathcal{K}_{ij}^{\tilde{t}}} q_{k'} / u_{ij} = \sum_{\tilde{t} \in \mathcal{T} : \mathcal{K}_{ij}^{\tilde{t}} \neq \emptyset} s_{C_{ij}^{q\tilde{t}}} w_{C_{ij}^{q\tilde{t}}}$ and constraints (9) are satisfied. Set the value of all $(\gamma^{Cons}, w^{Cons}, y^{Cons})$ variables not already given a value to zero. The resulting values $(\gamma^{Cons}, w^{Cons}, y^{Cons})$ are feasible for $Cons-SNSP(\mathcal{C})$ and have objective function value $\sum_{(i,j) \in \mathcal{A}} f_{ij} y_{ij}^{Cons} = \sum_{(i,j) \in \mathcal{A}} f_{ij} \sum_{((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}}} y_{ij}^{tt'TEN}$.

In the second part of the proof, let us consider a solution $(\gamma^{Cons}, w^{Cons}, y^{Cons})$ to $Cons-SNSP(\mathcal{C})$. We show that it can be transformed to a feasible solution of the $TEN-SNSP(\mathcal{D}_{\mathcal{T}})$. Consider a given $k \in \mathcal{K}$ and dispatch variable values $\gamma_{v_i^k v_{i+1}^k}^{kCons}$. Let $t = \gamma_{v_i^k v_{i+1}^k}^{kCons}$ and for all $i = 1, \dots, r_k - 1$ find the smallest t' such that $((v_i^k, t), (v_{i+1}^k, t')) \in \mathcal{A}_{\mathcal{T}}$. Note such a t' must exist by construction of $\mathcal{D}_{\mathcal{T}}$ and because the variable values γ^{Cons} satisfy constraints (11) and (12) of $Cons-SNSP(\mathcal{C})$. Set $x_{v_i^k v_{i+1}^k}^{ktt'TEN} = 1$. Note that when $\gamma_{v_i^k v_{i+1}^k}^{kCons} + \tau_{v_i^k v_{i+1}^k} < \gamma_{v_{i+1}^k v_{i+2}^k}^{kCons}$ the values of some variables $x_{v_{i+1}^k v_{i+2}^k}^{ktt+1TEN}$ associated with waiting arcs $((v_{i+1}^k, t), (v_{i+2}^k, t+1)) \in \mathcal{H}_{\mathcal{T}}$ will have to be set to 1 as well. A similar statement holds when $\gamma_{v_{r_k-1}^k v_{r_k}^k}^{kCons} + \tau_{v_{r_k-1}^k v_{r_k}^k} < d_k$. Set all other values $x_{ij}^{ktt'} = 0$. The resulting values x^{TEN} satisfy the flow balance constraints (2) and domain constraints (4) in $TEN-SNSP(\mathcal{D}_{\mathcal{T}})$.

Next, consider an arc $(i, j) \in \mathcal{A}$ and consolidation $C_{ij} \in \mathcal{C}_{ij}$ such that $w_{C_{ij}}^{Cons} = 1$. Let $t = \gamma_{C_{ij}}^k$ for some $k \in C_{ij}$. Note that, because the variable values γ^{Cons}, w^{Cons} satisfy constraints (10), we have $x_{ij}^{ktt'TEN} = 1 \quad \forall k \in C_{ij}$ by construction. Let $\mathcal{C}_{ij}(t)$ denote the set of consolidations $C_{ij} \in \mathcal{C}_{ij}$ such that $w_{C_{ij}}^{Cons} = 1$ and $\gamma_{C_{ij}}^k = t \quad \forall k \in C_{ij}$. Let $y_{ij}^{tt'TEN} = \sum_{C_{ij} \in \mathcal{C}_{ij}(t)} s_{C_{ij}}$ for the smallest t' such that $((i, t), (j, t')) \in \mathcal{A}_{\mathcal{T}}$. Such a t' must exist by construction of $\mathcal{D}_{\mathcal{T}}$. The values $y_{ij}^{tt'TEN}$ satisfy the domain constraints (5) of $TEN-SNSP(\mathcal{D}_{\mathcal{T}})$. Recalling that $s_{C_{ij}} = \lceil \sum_{k \in C_{ij}} q_k / u_{ij} \rceil$, we have that $u_{ij} y_{ij}^{tt'TEN} = u_{ij} \sum_{C_{ij} \in \mathcal{C}_{ij}(t)} s_{C_{ij}} \geq \sum_{C_{ij} \in \mathcal{C}_{ij}(t)} \sum_{k \in C_{ij}} q_k x_{ij}^{ktt'TEN}$ and the values (x^{TEN}, y^{TEN}) satisfy constraints (3). Set all variables $y_{ij}^{tt'TEN}$ not already given a value to zero. The resulting variable values (x^{TEN}, y^{TEN}) are feasible for $TEN-SNSP(\mathcal{D}_{\mathcal{T}})$ with objective value $\sum_{((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}}} f_{ij} y_{ij}^{tt'TEN} = \sum_{(i,j) \in \mathcal{A}} f_{ij} \sum_{t \in \mathcal{T}} \sum_{C_{ij} \in \mathcal{C}_{ij}(t)} s_{C_{ij}} = \sum_{(i,j) \in \mathcal{A}} f_{ij} y_{ij}^{Cons}$.

We have that any feasible solution to $TEN-SNSP(\mathcal{D}_{\mathcal{T}})$ can be converted to a feasible solution of $Cons-SNSP(\mathcal{C})$ with the same objective value and vice versa. Thus, $z_{TEN}(\mathcal{D}_{\mathcal{T}}) = z_{Cons}(\mathcal{C})$. \square

□

B *Cons-SNSP*(\mathcal{C}) formulation for Example 1

Given the consolidations enumerated in Table 4, the *Cons-SNSP*(\mathcal{C}) formulation for Example 1 is below.

$$z_{Cons}(\mathcal{C}) = \text{minimize } f_{ab}y_{ab} + f_{bc}y_{bc} + f_{cd}y_{cd}$$

subject to

$$\begin{aligned}
& w_{C_{ab}^1} + w_{C_{ab}^3} = 1, w_{C_{ab}^2} + w_{C_{ab}^3} = 1, \\
& w_{C_{bc}^1} + w_{C_{bc}^4} = 1, w_{C_{bc}^2} + w_{C_{bc}^5} = 1, w_{C_{bc}^3} + w_{C_{bc}^4} = 1, \\
& w_{C_{cd}^1} + w_{C_{cd}^3} = 1, w_{C_{cd}^2} + w_{C_{cd}^3} = 1, \\
& w_{C_{ab}^1} + w_{C_{ab}^2} + 2w_{C_{ab}^3} \leq y_{ab}, \\
& w_{C_{bc}^1} + w_{C_{bc}^2} + w_{C_{bc}^3} + 2w_{C_{bc}^4} + w_{C_{bc}^5} \leq y_{bc}, \\
& w_{C_{cd}^1} + w_{C_{cd}^2} + w_{C_{cd}^3} \leq y_{cd}, \\
& \gamma_{ab}^1 - \gamma_{ab}^3 \leq 1(1 - w_{C_{ab}^3}), \gamma_{ab}^3 - \gamma_{ab}^1 \leq 4(1 - w_{C_{ab}^3}), \\
& \gamma_{bc}^1 - \gamma_{bc}^3 \leq 2(1 - w_{C_{bc}^4}), \gamma_{bc}^3 - \gamma_{bc}^1 \leq 3(1 - w_{C_{bc}^4}), \\
& \gamma_{bc}^2 - \gamma_{bc}^3 \leq 3(1 - w_{C_{bc}^5}), \gamma_{bc}^3 - \gamma_{bc}^2 \leq 1(1 - w_{C_{bc}^5}), \\
& \gamma_{cd}^2 - \gamma_{cd}^3 \leq 3(1 - w_{C_{bc}^3}), \gamma_{cd}^3 - \gamma_{cd}^2 \leq 1(1 - w_{C_{cd}^3}), \\
& \gamma_{ab}^1 + 1 \leq \gamma_{bc}^1, \gamma_{bc}^2 + 1 \leq \gamma_{cd}^2, \\
& \gamma_{ab}^3 + 1 \leq \gamma_{bc}^3, \gamma_{bc}^3 + 1 \leq \gamma_{cd}^3, \\
& 1 \leq \gamma_{ab}^1 \leq 2, 2 \leq \gamma_{bc}^1 \leq 4, \\
& 4 \leq \gamma_{bc}^2 \leq 5, 5 \leq \gamma_{cd}^2 \leq 6, \\
& 1 \leq \gamma_{ab}^3 \leq 4, 2 \leq \gamma_{bc}^3 \leq 5, 3 \leq \gamma_{cd}^3 \leq 6, \\
& w_{C_{ab}^1}, w_{C_{ab}^2}, w_{C_{ab}^3}, w_{C_{bc}^1}, w_{C_{bc}^2}, w_{C_{bc}^3}, w_{C_{bc}^4}, w_{C_{bc}^5}, w_{C_{cd}^1}, w_{C_{cd}^2}, w_{C_{cd}^3} \in \{0, 1\}, y_{ab}, y_{bc}, y_{cd} \in \mathcal{Z}.
\end{aligned}$$