The Service Network Scheduling Problem

M. Hewitt[a,*], F. Lehuédé[b]

[a] Loyola University Chicago,Chicago,United States of America,mhewitt3@luc.edu
[b] IMT Atlantique, Nantes, France,fabien.lehuede@imt-atlantique.fr

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 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.

December 18, 2021

Keywords: Service Network Design Problem, Scheduling, Column Generation, Integer Programming

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 freight and small parcel industries. As transportation exhibits economies of scale, profitability for such companies 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.

More precisely, a *path* for a shipment refers to the sequence of terminals it visits, with the sequence beginning at the origin terminal for the shipment and ending at its destination terminal. We also refer to these paths as consisting of one or more transportation moves, where a *transportation move* refers to physical transportation, either by a shipment or a capacitated vehicle, between terminals in the network. Transportation moves executed by a vehicle incur a cost that is independent of the shipments it transports. Lastly, the *schedule* for a path prescribes dispatch times for each of its transportation moves. These schedules in turn imply vehicle dispatch times on those same moves. Achieving high levels of consolidation, 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. Jointly determining these paths and schedules is typically seen as a tactical planning problem and is modeled as a variant of the *Scheduled Service Network Design Problem* (SSNDP) (Crainic *et al.* , 2021).

A carrier may seek to keep shipment paths constant over some planning horizon (e.g. a month), in order to maintain consistency with the operations that support the execution of those paths. However, for some types of carriers (e.g. road-based Less-than-truckload freight transportation carriers) there is often flexibility, operationally-speaking, with respect to the scheduling of paths. In the near-term when more accurate forecasts of shipment volumes are known, it may be willing to adjust the schedules for those paths if doing so leads to greater 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).

We propose two mixed integer programming formulations of this problem. The first is based on the classical idea of a time-expanded network (Ford & Fulkerson, 1958, 1962). The second is based on enumerations of consolidations of shipments. 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 optimization problem to determine variables to add. It differs from Branch-and-Price in that the optimization problem solved is an integer program that is formulated so as to be a relaxation of the original problem. We refer to this algorithm as $IP - ColGen$, or, Integer Programming-based Column Generation. We prove the correctness of this algorithm and demonstrate with a computational study its superiority to solving the static formulation. We believe this algorithm and formulation technique can be employed in formulations of and solution methods for the more general SSNDP, something we will discuss during the presentation.

2 Methodology

In this section, we present the consolidation-based methodology for this problem. We omit the time-expanded network formulation for brevity and because it is classical. We model the terminal network and moves within that network with the directed network $\mathcal{D} = (\mathcal{N}, \mathcal{A})$, 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. 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 model shipments that must be routed through the network with the set $\mathcal{K}$ of commodities. 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 a size q_k quoted in the same unit as the capacity factors $u_{ij}, (i, j) \in \mathcal{A}$ associated with arcs.

Recalling that our problem presumes that paths for shipments are known, we let $p_k = v^k_1, ..., v^k_{r_k}$ represent the sequence of nodes from $\mathcal{N}$ in the path of commodity $k \in \mathcal{K}$, with $v^k_1 = o_k$ and $v^k_{r_k} = d_k$. We let $\mathcal{P} = \cup_{k \in \mathcal{K}} p_k$ denote the set of all such shipment paths. Relatedly, for commodity k we let the node set $\mathcal{N}^k \subseteq \mathcal{N}$ contain the nodes v^k_i in that path and the arc set $\mathcal{A}^k \subseteq \mathcal{A}$ contain the arcs (v^k_i, v^k_{i+1}) 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 $[\alpha^k_v, \beta^k_v]$ during which k can be at each node v in its path.

The proposed formulation of the SNSP that is based on sets of possible consolidations on each arc, $(i, j) \in \mathcal{A}$. To define the formulation we let $\mathcal{C}_{ij} = \{C^1, \ldots, C^{n_{ij}}\}$, $C^g \subseteq \mathcal{K}_{ij}\ \forall g = 1, \ldots, n_{ij}$, 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 $C^g = \{k_1^g, \ldots, k_{m_g}^g\} \in \mathcal{C}_{ij}$ $\forall g = 1, \ldots, n_{ij}$ if and only if $\cap_{q=1}^{m_g}[\alpha_i^{k_q}, \beta_i^{k_q}] \neq \emptyset$. 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 $g \in [1, n_{ij}]$ 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 $\{k_1^g, \ldots, k_{m_g}^g\}$ 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 to transport consolidation C.

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 $\gamma^k_{v_i^k v_{i+1}^k}, i = 1, \ldots, n_k - 1$ prescribe the time at which commodity k dispatches on the arc $(v_i^k, v_{i+1}^k), i = 1, \ldots, n_k - 1$ on its path. With these sets and decision variables, we define the optimization problem *Cons-SNSP*($\mathcal{C}$), as

$$z_{Cons}(\mathcal{C}) = \text{minimize} \sum_{(i,j)\in\mathcal{A}} f_{ij} y_{ij} \tag{1}$$

subject to

$$\sum_{C\in\mathcal{C}_{ij}} \phi_C^k w_C = 1 \qquad \forall k \in \mathcal{K}, (i,j) \in \mathcal{A}^k, \tag{2}$$

$$\sum_{C\in\mathcal{C}_{ij}} s_C w_C \leq y_{ij} \qquad \forall (i,j) \in \mathcal{A}, \tag{3}$$

$$\gamma_{ij}^k - \gamma_{ij}^{k'} \leq M_i^{kk'}(1 - \sum_{C\in\mathcal{C}_{ij}} \phi_C^k \phi_C^{k'} w_C) \qquad \forall (i,j) \in \mathcal{A}, k, k' \in \mathcal{K}_{ij}, \tag{4}$$

$$\gamma^k_{v_i^k v_{i+1}^k} + \tau_{v_i^k v_{i+1}^k} \leq \gamma_{v_{i+1}^k v_{i+2}^k} \qquad \forall k \in \mathcal{K}, i = 1, \ldots, n_k - 2, \tag{5}$$

$$\alpha_{v_i^k} \leq \gamma^k_{v_i^k v_{i+1}^k} \leq \beta_{v_i^k} \qquad \forall k \in \mathcal{K}, i = 1, \ldots, n_k - 1, \tag{6}$$

$$\gamma^k_{v_i^k v_{i+1}^k} \in \mathbb{N} \qquad \forall (v_i^k, v_{i+1}^k) \in p_k, k \in \mathcal{K}, \tag{7}$$

$$w_C \in \{0,1\} \qquad \forall C \in \mathcal{C}, \tag{8}$$

$$y_{ij} \in \mathbb{N} \qquad \forall (i,j) \in \mathcal{A}. \tag{9}$$

The objective seeks to minimize the total costs associated with vehicle moves that transport consolidations of shipments. Constraints (2) ensure that a consolidation is chosen for each commodity on each arc in its path. Constraints (3) ensure that sufficient capacity is paid for on each arc to support the consolidations chosen for that arc. Constraints (4) ensure that all commodities in the consolidation chosen for an arc dispatch at the same time. Constraints (5) ensure that the dispatch times for arcs on the path of a commodity agree with their travel times. Constraints (6) ensure the dispatch time decision variables occur within the corresponding time windows. Constraints (7), (8), and (9) define the decision variables and their domains. While the big-M value in constraints (4) may lead to a weak formulation, the formulation can be tightened without rendering any solutions infeasible by setting $M_i^{kk'} = \beta_i^{k'} - \alpha_i^k$.

3 Results

To compare the two formulations computationally, 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 other instance parameters.

We compare solving the consolidation-based formulation *Cons-SNSP*($\mathcal{C}$) with solving three variants of the time-expanded network formulation. The first, labeled "T-E Network" consists of solving the time-expanded network formulation over a complete time-expanded network. The second, labeled "Reduced T-E Network" consists of solving the formulation over a reduced time-expanded network. The third, labeled "Reduced T-E Network + Valid inequalities" consists of solving the formulation over a reduced time-expanded network but strengthened with valid inequalities.

Table 1 – *% solved, time to termination, averaged over all instances*

Method	% Solved	Time (sec.)
T-E Network	44.44%	4,094.54
Reduced T-E network	58.33%	3,080.16
Reduced T-E Network + Valid inequalities	66.67%	2,531.47
Consolidation	77.78%	1,770.94

However, a downside of the consolidation-based formulation is its enumerative nature. Thus, we have also developed a column generation-based algorithm to solve this formulation wherein consolidations are generated dynamically during the course of its execution. We will present this algorithm in detail during the talk. However, in Table 2 we report results from executing this algorithm on the same instances discussed above. We see that the algorithm dramatially

Table 2 – *Comparison of $IP - ColGen$ with Cons-SNSP*($\mathcal{C}$) *by number of commodities*

# Commodities	100	150	200	250	300	350	400	450	500	Average
IP-ColGen										
% Solved	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Time (sec.)	0.52	1.10	1.58	2.79	6.52	8.16	28.56	75.16	131.18	28.40
Cons-SNSP($\mathcal{C}$)										
% Solved	100.00%	100.00%	100.00%	100.00%	75.00%	75.00%	100.00%	25.00%	25.00%	77.78%
Time (sec.)	0.52	1.41	3.00	5.43	1,807.23	1,857.88	606.80	5,462.01	6,194.21	1,770.94

outperforms solving the static, *a priori* formulation.

References

Barnhart, Cynthia, Johnson, Ellis L, Nemhauser, George L, Savelsbergh, Martin WP, & Vance, Pamela H. 1998. Branch-and-price: Column generation for solving huge integer programs. *Operations research*, **46**(3), 316–329.

Crainic, Teodor Gabriel, Gendreau, Michel, & Gendron, Bernard. 2021. *Network Design with Applications to Transportation and Logistics.*

Desaulniers, Guy, Desrosiers, Jacques, & Solomon, Marius M. 2006. *Column generation.* Vol. 5. Springer Science & Business Media.

Ford, Lester Randolph, & Fulkerson, Delbert Ray. 1958. Constructing maximal dynamic flows from static flows. *Operations research*, **6**(3), 419–433.

Ford, Lester Randolph, & Fulkerson, Delbert Ray. 1962. *Flows in networks.* Princeton University Press.