CIRRELT

CIRRELT-2019-37

Scheduled Service Network Design with Resource Acquisition and Management under Uncertainty

Mike Hewitt
Teodor Gabriel Crainic
Maciek Nowak
Walter Rei

September 2019

Bureau de Montréal
Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél. : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec
Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél. : 1-418-656-2073
Télécopie : 1-418-656-2624

Scheduled Service Network Design with Resource Acquisition and Management under Uncertainty[ł]

Mike Hewitt[1,*], Teodor Gabriel Crainic[2], Maciek Nowak[1], Walter Rei[2]

[1] Department of Information Systems and Operations Management, Quinlan School of Business, Loyola University, 1 E. Pearson, Suite 204, Chicago, IL 60611, USA

[2] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Management and Technology, Université du Québec à Montréal, P.O. Box 8888, Station Centre-Ville, Montréal, Canada H3C 3P8

Abstract. We propose a scheduled service network design model that simultaneously addresses strategic decisions regarding fleet sizing and allocation, including acquisition and outsourcing, as well as tactical decisions regarding a repeatable transportation plan and schedule. Moreover, as a well-sized fleet and a well-designed transportation plan should be able to accommodate fluctuations in freight volumes, the model takes the form of a stochastic program, explicitly addressing uncertainty in demand through the use of scenarios. This is the first model to consider this full suite of decisions while also recognizing uncertainty in freight volumes. Given the computational difficulties associated with solving stochastic programs exactly, we propose a column-generation-based matheuristics scheme for addressing the model, which decomposes the optimization problem across multiple dimensions, and evaluates a neighboring solution across all scenarios. This is the first heuristic scheme for this class of problem and we assess its effectiveness on two sets of instances. The first is a set generated to mimic the operations of a Less-than-truckload freight transportation carrier and the second is based on the network of a European postal carrier. We see that the solution approach is able to produce high-quality solutions for both sets of instances in run-times that are acceptable in practice.

Keywords: Scheduled service network design, resource acquisition and management, stochastic programming, column generation, matheuristics.

Acknowledgments. Partial funding was provided by the Natural Sciences and Engineering Council of Canada (NSERC) through its Discovery Grants program. We gratefully acknowledge the support of Fonds de recherche du Québec through their strategic infrastructure grants.

[ł] Revised version of the CIRRELT-2017-52

Results and views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect those of CIRRELT.

Les résultats et opinions contenus dans cette publication ne reflètent pas nécessairement la position du CIRRELT et n'engagent pas sa responsabilité.

* Corresponding author: mhewitt3@luc.edu

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2019

1 Introduction

Consolidation carriers transport customer shipments that are small relative to vehicle capacity, enabling the exponential growth of world trade of consumer goods and the transformative effects of eCommerce. Consolidation is the basis of operations of less-than-truckload (LTL) motor carriers, railways moving both general and intermodal cargo, sea and river/canal intermodal navigation, small package/parcel courier companies, as well as the emerging business and organizational model for freight transportation, e.g., City Logistics and Physical Internet. To illustrate the importance of the industry, consider that in the U.S. alone, the revenue of the LTL industry was of the order of $34.9 billion in 2016 (Cassidy, 2017), while one player alone (UPS) in the small package/parcel industry reported $58 billion in revenue in 2018 (UPS, 2019). Consolidation carriers play a prominent role in the fulfillment of orders placed online, in brick-and-mortar stores, and through other channels. This research is focused on modeling and solving a planning problem faced by this type of freight transportation carrier.

For a consolidation carrier to deliver goods on time in a cost-effective manner, it must consolidate shipments, which in turn requires planning processes that coordinate the paths for different shipments in both space and time. These planning processes have long been assisted by solving the *Service Network Design* (SND) problem (Crainic, 2000; Wieberneit, 2008), which prescribes the choice of paths for shipments and the services or resources necessary to execute them. The main goal of the SND is to produce an operation (or load) plan that services demand while achieving the economic and service-quality targets of the carrier. Building such a plan involves selecting the services to operate, their schedules (departure times), and then routing customer shipments through the selected service network.

The set of potential plans that can be executed is directly impacted by the infrastructure a carrier has in place, including the terminals at which shipments can be handled, and the resources necessary for transportation. Associated with many types of resources (such as drivers) is a domicile or home terminal within the transportation network. And, for most resources, there are rules governing the movements they can make over a period of time. For example, for drivers, there are rules dictating that they must periodically return to their home terminal. As such, the feasibility and cost of executing a service network can be greatly impacted by the needs and rules that must be observed when managing resource usage. Fundamentally, carriers face the challenge of coordinating two different plans: (1) a plan for routing shipments that meets the service standards customers expect, and, (2) a plan for resource movements that observes governmental (and other) regulations.

Indeed, for many carriers, locating the resources that enable them to offer low-cost transportation services in a manner such that they are highly utilized is a pressing concern. Carriers in the United States have long listed driver shortage as one of their major concerns. And the severity of the shortage is often region-dependent, leading to great variation in the compensation of drivers that is primarily due to their home region. Another issue carriers face is that their fleet of vehicles can be heterogeneous with respect to fuel efficiency. While a portion of a carrier's fleet may be (relatively) new and fuel-efficient, there will still be vehicles that are older and have a higher cost per mile. The resulting challenge for a carrier is to have the most fuel-efficient vehicles in positions where they are readily available for the moves that involve the most miles.

Researchers have historically addressed the development of these plans separately, however. Such methods typically solve the service network design problem with no recognition of the need for resources to generate a set of transportation moves. Those moves are then used to instantiate a planning problem that seeks to cover the transportation activities with resources while observing rules regarding resource usage. Only recently have researchers proposed models and solution methods that recognize management issues related to resources (Andersen et al., 2009b; Crainic et al., 2014b). Recently, Crainic et al. (2017) proposed a model that links these two levels of decision-making: (1) strategic, wherein fleet sizing and allocation decisions are made, and, (2) tactical, wherein transportation plans are designed and executed. With an extensive computational study that paper illustrates the value in making these decisions jointly. However, that model assumes customer demands are known with certainty.

In this paper, we propose a stochastic program that considers the decisions prescribed by the model presented in Crainic et al. (2017), albeit in a context wherein uncertainty in freight volumes is explicitly recognized through scenarios. On the one hand, recognizing this stochasticity offers an opportunity to model a richer set of decisions a carrier may take in order to respond to unforeseen fluctuations in freight volumes. On the other hand, modeling the opportunity for decisions to depend on scenario can dramatically increase the number of decision variables and constraints in the resulting stochastic program. To mitigate this, and in anticipation of using a column generation-based solution approach, we propose a formulation that routes shipments on paths. This is in contrast to the model presented in Crainic et al. (2017), which, because it was not scenario-based, could route shipments on arcs without a dramatic increase in instance size. As few stochastic scheduled service network design problems of even moderate size have been solved exactly in the literature, and the model we propose considers a richer set of decisions, solving the stochastic program we propose also necessitates a new algorithm. As such, we propose a matheuristic for solving the model.

Matheuristics are meta-heuristics that make explicit use of the mathematical formulation in some parts of the search. Such methods have (computationally) proven to be an effective solution approach for hard combinatorial optimization problems. Matheuristics have primarily been applied to deterministic problems wherein the values of problem parameters are known with certainty (see Hewitt et al., 2010; Erera et al., 2013; Archetti et al., 2008; Schmid et al., 2009; Villegas et al., 2013; Archetti et al., 2015, for some examples). However, more and more researchers and practitioners are proposing and attempting to solve stochastic models, wherein only a distribution is known for some problem parameter values. This trend can be partially attributed to the increased availability of "big data" that has made possible the development of representative distributions for problem parameter values. It can also partially be attributed to the improved performance of commercial optimization solvers.

We believe this research makes multiple contributions. It presents the first stochastic programming model to help transportation companies size and allocate their fleet, while recognizing the impact of those decisions on operational (transportation) costs in different scenarios. This model is adaptable to different operational settings, including rules that must be followed when planning transportation schedules as well as the opportunity to use different transportation modes (e.g., truck vs. rail). We also validate the use of this model by illustrating the value of explicitly recognizing uncertainty in freight volumes. This research also presents a computationally effective matheuristic, which decomposes the optimization problem across multiple dimensions, and evaluates a neighboring solution across all scenarios. This is the first heuristic scheme for this class of problems and computational experiments on a set of generated instances suggest that it is effective at producing high-quality solutions. Lastly, we assess the effectiveness of the heuristic on instances derived from the operations of a European postal carrier and again see that it is effective at producing high-quality solutions.

The paper is organized as follows. In Section 2 we describe the problem studied in detail. A brief literature review of relevant service network design is given in Section 3. We then present the formulation, in Section 4, while Section 5 is dedicated to the solution method. We describe the experimental design in Section 6. In Section 7 we present our computational analysis of the value of modeling uncertainty. In Section 8 we present an extensive analysis of the proposed solution approach on the generated set of instances, while Section 9 presents results derived from applying the solution approach to "real-world" instances. We conclude in Section 10.

2 Problem statement

We study and solve a problem that spans strategic and tactical planning decisions made by a consolidation based carrier. Such a carrier transports freight through a network of terminals on what are often referred to as "services." As a result, the tactical decisions correspond to selecting the services to operate and determining how freight is routed through the resulting service network. However, these services and routes must also be supported by resources, which are in turn assigned to terminals.

The strategic decisions determine the total number of such resources available, as well as the assignment of each resource to a home terminal. Costs associated with these strategic decisions can include the purchase cost of a capital asset, salary and/or signing bonus associated with hiring an individual, and transportation costs associated with reallocating a resource from one home terminal to another. These strategic and tactical decisions can be (and often are) made independently. However, as shown in Crainic et al. (2017), doing so can lead to a significant increase in total costs. Thus, we seek to make them jointly.

While Crainic et al. (2017) studied a similar problem, that problem only considered a point forecast of customer demands when making decisions. Yet, unanticipated fluctuations in freight volumes can render plans made at both the strategic and tactical levels ineffective and costly. See Lium et al. (2009) for an analysis of how tactical service selection decisions should be made in the presence of uncertainty in freight volumes. Thus, we seek to make these strategic and tactical decisions jointly, and with an explicit recognition of uncertainty in freight volumes. In doing so, the strategic resource acquisition and allocation decisions can be made with an accurate estimate of their impact on the transportation costs the carrier will ultimately incur.

This estimate will be based on the costs associated with operating services and consolidation terminals to transport customer demands during a representative period of time, which is referred to as the *schedule length*. During this period of time, the carrier must transport a set of customer shipments for which the geographic (origin and destination locations) and temporal (availability time and due arrival time) attributes are assumed to be known. However, to ensure decisions (both strategic and tactical) that are robust with respect to customer demand volumes, only probability distributions for freight volumes are presumed as known. The resulting transportation plan, including the selected services and resource usages, will be repeatedly executed during the *tactical planning horizon*. As an example, a carrier may forecast that for the coming quarter shipment volumes can vary from one week to the next, but the underlying statistical distributions describing those variations are the same. In such a setting, the carrier may wish to derive a single, weekly transportation plan that will be executed each week in the quarter and is able to efficiently accommodate variations in shipment volumes from one week to the next.

At consolidation terminals in the network, shipments are sorted and consolidated into vehicles that will thus contain shipments from multiple customers, with different shipments potentially having different origins or destinations or both. Direct services (no intermediate stops) connect these terminals and specify how vehicles move. We note that in this paper we assume a homogeneous fleet of vehicles. However, the model we propose can be adapted to heterogeneous fleets. A *service* is thus defined by an origin terminal and a destination terminal, as well as by the time window during which the service will depart from the origin terminal and the time window during which it will arrive at the destination terminal. For modes such as rail, a service may only be executed at most once. For modes such as trucking, the same service may be executed multiple times (e.g. multiple trucks can dispatch from a terminal at the same time). Resources are associated with these services and provide the means to perform them. Each resource must be assigned to a "home" terminal. We consider a setting wherein each resource operates according to a cyclic route that starts at and returns to its home terminal and enables it to support services.

Shipments will thus be routed through the service network enabled by resources, being sorted and consolidated at each intermediary terminal on this route. Shipment routing also displays a temporal component as the carrier has to decide when a shipment should dispatch from each terminal on its route. Indeed, shipments may be held at a terminal for a later-departing service to allow for consolidation with later-arriving shipments. Of course the decision to hold a shipment to achieve greater consolidation must be balanced against the need to deliver the shipment at the time the customer expects. There are various costs associated with executing a service, including costs associated with terminal operations that support the service and the transportation itself. Similarly, there are costs associated with handling a shipment at a terminal. Section 4 elaborates on these with the model description.

While resources are necessary to execute services, we do not require that a shipment needs to be assigned to a single resource for transportation from its origin to its destination. As such, a shipment may be transferred from one resource to another when travelling on a sequence of services, with each of those services supported by a different resource. The rules governing the movements a resource may make during the schedule length can be complex and depend upon what the resource is; for a human resource (say a driver) government agencies (such as the United States Department of Transportation) specify many limits (sometimes called "hours-of-service" regulations) upon what they may do (FMCSA, 2014). For example, a driver may drive at most 11 hours a day after 10 consecutive hours off duty (at rest).

This paper utilizes a simple set of rules governing what movements a resource may make. Specifically, we presume that the resource must return to its home terminal at least once during the schedule length. However, the proposed model and solution method can be easily adapted to other cases. The sequence of movements made by a resource during the schedule length is called an *itinerary.* There are also costs associated with using a resource from a specific home terminal, such as those incurred due to maintenance.

We also consider the option that a service be supported not by a resource owned (or leased) by the transportation company, but instead by a third party. In this situation, the resource is "acquired" from the third party only for the execution of this service and the carrier itself need not ensure that the resource's complete itinerary (which may include moves for other carriers) follows the appropriate rules. Outsourcing a service to a third party-owned resource is presumed to incur costs that are greater than executing the service with an owned resource. The problem studied in this paper considers the decisions prescribed by the model proposed in Crainic et al. (2017). However, explicitly recognizing uncertainty in freight volumes provides a context wherein a richer set of decisions can be considered. Specifically, we model opportunities for outsourcing transportation activities in response to observing actual freight volumes.

To precisely define the decision-making process, we next identify which decisions are made before volumes are known, and which after (i.e., the recourse). We presume that decisions related to resource acquisition and allocation, as well as those regarding which services to execute, are made before volumes are known. Similarly, we assume that decisions regarding which services to outsource can be made before freight volumes are known. In this case, we consider a situation wherein a carrier signs a long-term contract with another carrier. Fundamentally, we presume that the planner must make capacity decisions, including the size and allocation of a fleet of resources and a repeatable transportation plan that uses those resources, before freight volumes are known.

However, we also presume that the carrier can take two recourse actions in response to observing freight volumes: (1) routing an individual shipment from its origin to its destination based on all shipment volumes and given available capacity (which may involve using a third party carrier), and, (2) to outsource the execution of a service. With this second recourse, we are considering a situation wherein a carrier contracts a third party transportation provider on the “spot” market. The costs associated with outsourcing via the “spot” market are presumed to be higher than via long-term contract.

Ultimately, the problem the planner faces is to determine the number and allocation of resources that best balance the extra costs associated with resource acquisition and re-allocation with the transportation savings these decisions enable. Of course, balancing these costs also necessitates putting them on the same scale; for a schedule length of a week (or even a month) the transportation savings realized by purchasing a new truck will rarely outweigh its purchase cost. As such, when defining this problem, we assume that the acquisition and re-allocation costs are amortized or spread out over periods of time longer than the length of the schedule length.

3 Literature review

The planning problem studied here links two types of decisions: determining the acquisition and allocation of resources and how to transport customer shipments, whose volume is not known with certainty, using those resources. The resource acquisition and allocation decisions can be seen as facility location-type decisions, whereas determining how to transport customer shipments whose volumes are not known can be viewed as stochastic service network design-type decisions. As such, we next review related literature in the facility location and service network design domain. At the same time this problem locates resources that must be managed, and thus this section concludes with a review of the literature on service network design problems that recognize the need for resources and how they must be managed.

We first refer the reader to the review of Contreras and Fernandez (2012), which provides a unified view of problems that combine location and network design issues. Melkote and Daskin (2001b) present an optimization model that chooses locations for (uncapacitated) facilities as well as designs a transportation network based on those facilities. Building off this work, Melkote and Daskin (2001a) introduce a combined facility location/capacitated network design problem in which facilities have capacities on the amount of customer shipment demand they can serve. However, neither model captures the resources that are needed to support the transportation network. Similar to this work, Crainic et al. (2017) jointly model resource acquisition, allocation, and management decisions along with decisions regarding the design of a transportation network. However, all of the above works presume that commodity volumes are known with certainty.

We next consider the literature on service network design problems that recognize the need for and management of resources. Early papers (Kim et al., 1999; Smilowitz et al., 2003; Lai and Lo, 2004) studied problems modeling the requirement that the number of services entering and leaving a terminal at a point in time must be equal. These models assume one type of resource and that each service is supported by one unit of that resource. As a result, this constraint (often called *design-balance* constraints, Pedersen et al., 2009) ensures a balance of resources at each terminal and point in time. Similar types of constraints can be found in papers wherein the resource modeled is a container (Powell, 1986; Jarrah et al., 2009; Erera et al., 2013).

Regarding solution methods, Pedersen et al. (2009) observed that the addition of these design-balance constraints can complicate the search for high quality solutions as rounding-based techniques are likely to produce an infeasible solution. As a result, Pedersen et al. (2009) proposed a two-phase tabu-search method wherein the first phase explores the space of solutions that satisfy flow constraints but not necessarily design-balance constraints. The second phase is entered when a solution from the first does not satisfy the design-balance constraints, wherein a path-based neighborhood heuristic is used to convert the solution to one that is feasible for the full problem. However, the quality of the solution depends heavily on this second phase, which they observed required a significant number of iterations to produce a feasible solution.

Following up on that work, Vu et al. (2013) proposed an approach that can efficiently convert an infeasible solution (which satisfies the flow constraints but not design-balance constraints) to a feasible one using a minimum cost maximum flow procedure. The procedure is integrated into a three-phase matheuristic which combines tabu-search, path-relinking and exact optimization and this solution approach was found to be effective at finding high-quality solutions in reasonable run-times. In addition, this minimum cost maximum flow model was also used (Crainic et al., 2014b) in a solution method for another service network design problem that models resource constraints and was effective in that setting as well. Simultaneously, Chouman and Crainic (2015) proposed a competitive matheuristic based on a cutting plane approach which was able to produce high quality solutions in short running times.

The design-balance constraints naturally imply a cycle-based formulation. As such, Andersen et al. (2009a) compared cycle and arc-based formulations and observed that the use of cycle-based formulations enabled a more effective search for high quality primal solutions and yielded stronger dual bounds. As a result, Andersen et al. (2011) presented a cycle-based branch-and-price solution method to solve this problem for moderate instance dimensions.

However, these cycle-based formulations were used not as a modeling tool but rather for their impact on algorithmic effectiveness. Crainic et al. (2014b) instead used a cycle-based formulation to model a limit on how many resources are available at each terminal and that there are rules regarding what a resource may do during the planning horizon. The authors present a solution approach for this problem that combines column generation, slope-scaling, and exact optimization, together with an extensive computational study illustrating its effectiveness.

As this literature review highlights, there are few studies addressing service network design with resource management concerns, with only a handful aiming to combine strategic and tactical decisions. There has been work on service network design models that explicitly recognize uncertainty (Crainic et al., 2011, 2014a; Lium et al., 2009; Lai and Lo, 2004; An and Lo, 2014), but none of these models consider resource acquisition and deployment decisions. To the best of our knowledge, no literature considers demand uncertainty and how it affects both the design of the service network and the acquisition and deployment of resources to support that network. Our contribution to addressing this issue follows.

4 Optimization Model

The proposed model is based on the premise that each customer shipment has a known origin and destination, but that there is uncertainty regarding its volume. This uncertainty is incorporated into the decision-making process through the use of a two-stage, scenario-based model. This model is similar to traditional stochastic network design models (Lium et al., 2009; Crainic et al., 2011, 2014a), in that tactical decisions regarding itineraries for resources and the associated service network are first stage decisions and only shipment routes and outsourcing decisions can be made after shipment volumes are revealed. In the model we present, the first stage also includes strategic decisions regarding the number of resources acquired, the allocation of new resources, and the re-allocation of existing ones. The second stage of the model provides an estimate of the impact of both the strategic and tactical first stage decisions on operating costs over multiple, repeating, periods of operation (e.g. weeks).

Representing uncertainty with scenarios in a mathematical program can have algorithmic implications, as modeling the opportunity for decisions to depend on scenario can dramatically increase the number of decision variables and constraints in the resulting stochastic program. In the problem we study, a set of shipment routing variables, along with constraints to ensure shipments depart from their origins and arrive at their destinations, must be created for each scenario. The impact of scenarios on model size is further amplified by the fact that we consider a scheduled service network design problem, and thus the shipment routing variables we consider must also be indexed by time in some manner. As such, to manage model size, we model the routing of shipments with path variables, which will be generated dynamically. Lastly, we note that as this work extends that of Crainic et al. (2017), which considers a determinstic variant of this problem, some of the model definitions are drawn from that paper.

We first discuss the modeling of tactical decision-making. We assume the carrier transports shipments through a physical network of terminals, represented by the set Λ. Services are used to transport shipments, with Σ denoting the set of potential direct services between terminals in Λ, which the model will select and schedule to be included in the plan. In a practical implementation of this model, different terminals may have different capabilities (e.g., ports vs. ground terminals), and different services may represent different modes. However, for simplicity of presentation, we will not make any presumptions or discuss details regarding the infrastructure, roads or rail tracks, over which transportation is performed.

At the tactical level of planning, services are selected and scheduled over the schedule length, which is divided into $\mathcal{T} = \{1, 2, \ldots, TMAX\}$ time periods. The selected plan will then be repeated on a schedule-length basis. Based upon these periods, we create a time-space network, $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, a directed graph that models transportation activities at different points in time with different nodes and arcs. Specifically, the node set $\mathcal{N}$ models the operations of terminals in different periods, i.e., $\mathcal{N} = \Lambda \times \mathcal{T} = \{l_t | l \in \Lambda, t \in \mathcal{T}\}$, where l_t represents terminal l at period t. The arc set $\mathcal{A}$ contains two types of arcs. The first is a *service arc* (from the set Σ) and models the operation of a service between two terminals at a particular point in time. The second is a *holding arc* and models the opportunity for a resource or shipment to idle at a terminal from one period to the next. We denote the set of service arcs by $\mathcal{S}$ and holding arcs by $\mathcal{H}$, and thus $\mathcal{A} = \mathcal{S} \cup \mathcal{H}$.

Regarding service arcs, for each possible service $s = (l, m)$ between terminals $l, m \in \Lambda$ and time period $t \in \{1, \ldots, TMAX\}$, we add the arc $(l_t, m_{t'=(t+\pi_{lm}) \bmod TMAX})$ to $\mathcal{S}$

(assuming the service from l to m requires π_{lm} periods of travel time). Due to the presumption that freight demands follow a repetitive pattern, we construct the time-space network to support designing schedules that will be repeated. Specifically, we model a service of length π_{lm} that departs from a terminal in period t as arriving at the destination in period $(t + \pi_{lm})$ mod $TMAX$. With a limit of u_s on how much shipment demand can be carried by service $s = (l, m)$, we set the capacity, $u_{l_t m_{t'}}$, of executions of that service at different times t to u_s. Regarding the *holding arcs*, we add to $\mathcal{H}$ arcs of the form $(l_t, l_{(t+1) \text{mod } TMAX})$ for each terminal l and period t. While these arcs are assumed to be uncapacitated (both with respect to shipment demands and resources) in our experiments, terminal capacities (on shipments or resources) could be modeled by placing capacities on these arcs.

We model a shipment that is available in terminal l in period a and must be transported to terminal m by period b as a commodity with index k, origin node $o(k) = l_a$, and destination node $d(k) = m_b$. Thus, an individual commodity is defined by a combination of origin terminal (l), available time (a), destination terminal (m), and due time (b). In addition, when multiple shipments share this same combination, a commodity can represent an aggregation of those individual shipments. The set of all shipments is represented by $\mathcal{K}$. While the set of commodities, and their origin and destination cities and times, are presumed known, we model uncertainty in their volume with the set of scenarios, Ψ, with each scenario having a probability ψ of occurring. The value $q^{k\psi}$ represents the volume of commodity $k \in \mathcal{K}$ in scenario $\psi \in \Psi$, with $q_i^{k\psi} = q^{k\psi}$ when $i = o(k)$, $q_i^{k\psi} = -q^{k\psi}$ when $i = d(k)$, and $q_i^{k\psi} = 0$ otherwise.

Regarding the routing of a shipment, we consider for commodity $k \in \mathcal{K}$ and scenario $\psi \in \Psi$ a set of paths, $\mathcal{P}_k^{\psi}$, each of which constitutes a sequence of scheduled services (from the set $\mathcal{S}$) from that commodity's origin, $o(k)$, to its destination, $d(k)$. Each path has a cost c_p that corresponds to the total variable cost paid for the services in that path. Specifically, $c_p^k = \sum_{(l_t, m_{t'}) \in \mathcal{A} \cap p} c_{l_t m_{t'}}^k$, where $c_{l_t m_{t'}}^k$ is the cost of commodity k traveling on service $(l_t, m_{t'})$. For a service arc, this cost parameter can model handling costs associated with loading the shipment into a vehicle at the origin terminal and unloading at the destination terminal. This parameter can also model the impact the weight of a shipment can have on the cost of executing a service. For a holding arc, this parameter can model other handling activities, or, the allocation of the cost of physical space to shipments based on the amount of space in the terminal they require. The decision of how to route a shipment is made after demands are observed, and thus the continuous variable $x_p^{k\psi}$ represents the fraction of commodity k's demand that travels along path p in scenario ψ. For each commodity k, the set $\mathcal{P}_k^{\psi}$ contains one path that corresponds to direct delivery of that shipment from its origin to its destination by an external carrier.

For a commodity to travel on the service arc $(l_t, m_{t'}) \in \mathcal{S} \subset \mathcal{A}$, that service must be "executed," and the binary decision variable $y_{l_t m_{t'}}$ models this choice. Executing a service incurs a fixed cost, f_{lm}. This parameter can be used to model overhead costs, such as facility maintenance and labor, as well as the actual transportation cost of a resource traveling from terminal l to terminal m. This parameter can also be indexed by the time period t for settings wherein transportation costs are time-dependent, such as in areas where congestion-based traffic pricing is used. We also model the option of executing that same service, albeit with the use of a third party-owned resource. The binary variable $y^{e}_{l_t m_{t'}}$ models this choice, which incurs a different fixed cost, f^{e}_{lm}. For most practical settings, we anticipate this parameter value will be a function of the same overhead costs as those that contribute to the value of f_{lm}, as well as costs charged by the third party carrier. Finally, to model the second recourse, wherein a service is outsourced on a "spot" market after demand has been observed, we use the binary variable $y^{\sigma\psi}_{l_t m_{t'}}$ which incurs the fixed cost f^{σ}_{lm}.

We model that executing a service requires the use of a resource that must periodically return to its assigned home terminal. Similar to the research presented in Crainic et al. (2014b), a cycle, τ, models a sequence of possible movements during the schedule length for a resource assigned to terminal h in the graph $\mathcal{G}$ that begins and ends at node $h_t \in \mathcal{N}$ for some $t \in \mathcal{T}$. We denote the set of such cycles by θ_{h_t} and let $\theta_h = \cup_{t=1}^{TMAX} \theta_{h_t}$, the set of all cycles that require a resource assigned to terminal h and that depart from there at some time period during the schedule length. The rules governing the movements a resource may make during the schedule length are encoded in the definition of the set θ_{h_t}. Note that this allows the modeling of rules that vary both by the terminal h to which the resource is assigned and the period t during which the itinerary begins. For our experiments, we only impose the rule that the itinerary for a resource must begin and end at the resource's assigned terminal. Thus a valid cycle is one that begins by departing from h in period t and ends by arriving at l, albeit $TMAX$ periods later. Note that a cycle beginning at h_t may return to h multiple times, and if it last returns to h in period $t^{'} < t + TMAX$ appended holding arcs allow it to reach h_{t+TMAX}.

The binary variable $z^{\tau}_{h} \in \{0, 1\}$ represents whether a resource with home terminal $h \in \Lambda$ executes cycle $\tau \in \theta_h$. The parameter O^{τ}_{h} models the costs associated with this route, which can include maintenance. However, as τ also models the route traveled by the resource, the value of O^{τ}_{h} includes the corresponding transportation costs as well. Regarding the pairing of services with resource itineraries, let $r^{\tau}_{l_t m_{t'}}$ (binary) denote whether arc $(l_t, m_{t'}) \in \mathcal{A}$ is contained in cycle τ.

Regarding strategic decisions, a conceptual "source" layer in the time-space network is used to model resource acquisition and allocation decisions. There are two types of nodes in this layer. The first is an "Acquisition node," denoted by A, that represents the acquisition of a new resource. This node connects to each of the terminals l at the beginning of the tactical planning horizon with an arc that represents the allocation of a newly acquired resource to that terminal. The second type of node models the re-allocation of existing resources. As such, we add a node for each terminal $l^{'} \in \Lambda$ to this layer and then arcs connecting that node to each terminal, $l \in \Lambda$, at the beginning of the tactical planning horizon. These arcs represent the re-allocation of a resource currently assigned to terminal l to terminal l'. For simplicity when developing the mathematical model, we include arcs wherein $l = l'$, in which case the resource is not re-positioned. Finally, let Λ^+ denote the set of terminals, Λ, along with the Acquisition node. We illustrate this expanded network in Figure 1, wherein arc a between node A in the source layer and node T3 at time period 1 models the acquisition of a new resource that is allocated to T3. Similarly, arc b between T1 in the source layer and T2 at time period 1 models the re-allocation of a resource currently allocated to T1 to T2. Finally, arc c between T1 in the source layer and in time period 1 models a resource that remains at T1.

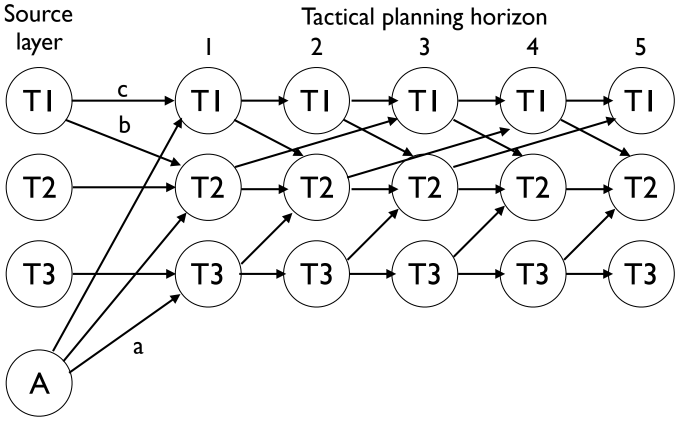

Figure 1: Modeling strategic and tactical decisions (not all arcs included)

The integer variable $a_{wh}, w \in \Lambda^+, h \in \Lambda$ then represents the number of resources acquired from source w (either through acquisition or repositioning) and assigned to terminal h. Assigning a resource to terminal h from source w has a cost F_{wh}. When w corresponds to the Acquisition node, the variable represents the purchase of a new resource and subsequent allocation to terminal h. As such, if the resource being modeled is equipment, F_{wh} could include the acquisition cost, only amortized. If the resource is an individual, this parameter could include wages and some amortization of a signing bonus paid to the individual. When w represents an existing terminal then the variable a_{wh} corresponds to the allocation of a resource that is currently assigned to terminal w to terminal h. In this case, F_{wh} includes any costs associated with such an action, such as transportation. When $w = h$, this variable represents leaving resources at their currently assigned terminal. Let I_w represent the number of existing resources assigned to terminal w.

We summarize the parameters and variables of the model in the following tables.

Ultimately, we seek to solve what we call the *Scheduled Service Network Design with Resource Acquisition and Management under Uncertainty* (*SSND-RAMU*) prob-

Table 1: Model parameters

Parameter	Meaning
F_{wh}	Cost of assigning a resource to terminal h from source w.
O^{τ}_{h}	Cost of operating cycle τ by a resource with home terminal h.
f_{lm}	Cost of executing service (l, m) with owned resources.
f^{e}_{lm}	Cost of outsourcing via long-term contract the execution of service (l, m).
ϕ_{ψ}	Probability of scenario ψ occurring.
c^{k}_{p}	Per-unit-of-volume cost of commodity $k \in \mathcal{K}$ traveling on path p.
$q^{k\psi}$	Volume of commodity k in scenario ψ.
f^{σ}_{lm}	Cost of outsourcing on a spot market in scenario ψ the execution of service (l, m).
$u_{l_t m_{t'}}$	Capacity of service (l, m) when dispatched at time period t.
I_w	Number of resources initially assigned to terminal w.
$r^{\tau}_{l_t m_{t'}}$	0/1 indicator of whether cycle τ contains service (l, m) dispatching at time period t.

Table 2: Model variables

Variable	Meaning
$a_{wh} \in Z$	Number of resources assigned to terminal h from source w.
$z^{\tau}_{h} \in \{0, 1\}$	Whether cycle τ is executed by a resource with home terminal h.
$y_{l_t m_{t'}} \in \{0, 1\}$	Whether service (l, m) dispatched at time period t is executed by owned resources.
$y^{e}_{l_t m_{t'}} \in \{0, 1\}$	Whether service (l, m) dispatched at time period t is outsourced via long-term contract.
$y^{\sigma}_{l_t m_{t'}} \in \{0, 1\}$	Whether service (l, m) dispatched at time period t is outsourced via spot market.
$x^{k\psi}_{p} \geq 0$	The fraction of commodity k's volume that travels on path p in scenario ψ.

lem, which in its scenario-based, deterministic form seeks to

$$\text{minimize} \sum_{w \in \Lambda^{+}} \sum_{h \in \Lambda} F_{wh} a_{wh} + \sum_{h \in \Lambda} \sum_{\tau \in \theta_h} O^{\tau}_{h} z^{\tau}_{h} + \sum_{(l_t, m_{t'}) \in \mathcal{S}} f_{lm} y_{l_t m_{t'}} + \sum_{(l_t, m_{t'}) \in \mathcal{S}} f^{e}_{lm} y^{e}_{l_t m_{t'}} + \tag{1}$$

$$+ \sum_{\psi \in \Psi} \phi_{\psi} * (\sum_{k \in \mathcal{K}} \sum_{p \in \mathcal{P}^{\psi}_{k}} c^{k}_{p} q^{k\psi} x^{k\psi}_{p} + \sum_{(l_t, m_{t'}) \in \mathcal{S}} f^{\sigma}_{lm} y^{\sigma\psi}_{l_t m_{t'}}) \tag{2}$$

subject to

$$\sum_{p \in \mathcal{P}^{\psi}_{k}} x^{k\psi}_{p} = 1 \quad \forall k \in \mathcal{K}, \psi \in \Psi, \tag{3}$$

$$\sum_{k \in \mathcal{K}} \sum_{p \in \mathcal{P}^{\psi}_{k} : (l_t, m_{t'}) \in p} q^{k\psi} x^{k\psi}_{p} \leq u_{l_t m_{t'}} (y_{l_t m_{t'}} + y^{e}_{l_t m_{t'}} + y^{\sigma\psi}_{l_t m_{t'}}), \quad \forall (l_t, m_{t'}) \in \mathcal{S}, \psi \in \Psi \tag{4}$$

$$\sum_{h \in \Lambda} a_{wh} = I_w, \quad \forall w \in \Lambda, \tag{5}$$

$$\sum_{\tau \in \theta_h} z^{\tau}_{h} = \sum_{w \in \Lambda^{+}} a_{wh}, \quad \forall h \in \Lambda, \tag{6}$$

$$y_{l_t m_{t'}} + y^{e}_{l_t m_{t'}} + y^{\sigma\psi}_{l_t m_{t'}} \leq 1, \quad \forall (l_t, m_{t'}) \in \mathcal{S}, \psi \in \Psi, \tag{7}$$

$$y_{l_t m_{t'}} \leq \sum_{h \in \Lambda} \sum_{\tau \in \theta_h} r^{\tau}_{l_t m_{t'}} z^{\tau}_{h}, \quad \forall (l_t, m_{t'}) \in \mathcal{S}, \tag{8}$$

$$a_{wh} \in Z, \quad \forall w \in \Lambda^{+}, h \in \Lambda, \tag{9}$$

$$z_h^\tau \in \{0,1\}, \quad \forall h \in \Lambda, \tau \in \theta_h, \tag{10}$$

$$y_{l_t m_{t'}} \in \{0,1\}, \quad \forall (l_t, m_{t'}) \in \mathcal{S}, \tag{11}$$

$$y^e_{l_t m_{t'}} \in \{0,1\}, \quad \forall (l_t, m_{t'}) \in \mathcal{S}, \tag{12}$$

$$y^{\sigma\psi}_{l_t m_{t'}} \in \{0,1\}, \quad \forall (l_t, m_{t'}) \in \mathcal{S}, \psi \in \Psi, \tag{13}$$

$$x_p^{k\psi} \geq 0, \quad \forall p \in \mathcal{P}_k^\psi, \, k \in \mathcal{K}, \psi \in \Psi. \tag{14}$$

The objective of this model consists of two components: (1) the costs that are incurred before demand is realized, term (1), and, (2): the expected recourse, term (2). The sum of these costs is then minimized. Constraints (3) ensure a path is chosen for each commodity in each scenario. Constraints (4) ensure that, in all scenarios, whenever a service is executed (either by an owned resource, an outsourced resource that was acquired via long-term contract, or an outsourced resource that was acquired on the spot market) its associated capacity is sufficient to flow the total amount of demand that is transported via the chosen paths that include the specific service. The remaining constraints in the model are the same as those from Crainic et al. (2017), albeit defined over the set of scenarios. Constraints (5) ensure that all resources initially assigned to a given terminal are allocated to a terminal (possibly the same) during the planning horizon. Constraints (6) then limit the number of cycles chosen that originate at a terminal by the number of resources assigned to that terminal (note that the right-hand-side of this constraint includes acquired resources). Constraints (7) ensure that each service is executed at most once. Constraints (8) ensure that services that are executed and require an owned resource are covered by an owned resource. Finally, constraints (14), (9), (10), (11), (12) and (13) define the decision variables of the model and their domains.

The model just presented can be adapted to different settings. As noted, constraints (7) are appropriate for transportation modes such as rail, wherein only one vehicle can be dispatched in a period. For modes where this is not the case, such as truck, constraints (7) can be removed. In addition, this model presumes a homogeneous fleet of vehicles. A heterogeneous fleet, wherein transportation costs and capacities depend on the vehicle used could be modeled by using service variables of the form $y^v_{l_t m_{t'}}$, service costs of the form f^v_{lm}, and capacity parameters of the form $u^v_{l_t m_{t'}}$ wherein v indicates the type of vehicle used. Similarly, resource-related decisions may depend on the industrial setting as well. In some, the relocation of resources may be challenging if not impossible. For those settings, the model may be restricted to only determine the acquisition and allocation of new resources by fixing the variables $a_{wh} = 0, w \in \Lambda$. In other settings, an organization may not want to increase the size of its fleet and is instead only interested in reallocation. This could be accomplished by fixing the variables $a_{Ah} = 0$, recalling that node A in the network represents acquisition.

5 Solution Approaches

The model we seek to solve consists of two sets of variables, paths and cycles, that are typically too large to be enumerated *a priori*. Thus, two solution approaches to the SSND-RAMU presented in this section generate these variables dynamically. The first solution approach is a column generation-based (CG; Bertsimas and Tsitsiklis, 1997) heuristic (Section 5.2), and serves as a benchmark for the second approach presented, a matheuristic (Section 5.3). As both heuristics rely on solving the linear programming relaxation, SSND-RAMU$_{LPR}$, of SSND-RAMU with a CG-based procedure, we start by describing that procedure.

5.1 Solving SSND-RAMU$_{LPR}$

Both proposed solution methods dynamically generate path, $x_p^{k\psi}$, and resource cycle variables z_h^τ. As such, we next describe a column generation-based procedure for solving the SSND-RAMU$_{LPR}$ that repeatedly generates variables of each type.

Given a set of commodity paths, $\bar{\mathcal{P}} = \cup_{\psi \in \Psi, k \in \mathcal{K}} \bar{\mathcal{P}}_k^\psi \subseteq \mathcal{P}_k^\psi$, and a set of resource cycles $\bar{\theta} \subseteq \theta = \cup_{h \in \Lambda} \theta_h$, the SSND-RAMU$(\bar{\theta}, \bar{\mathcal{P}})$ (and its linear programming relaxation SSND-RAMU$(\bar{\theta}, \bar{\mathcal{P}})_{LPR}$) is the SSND-RAMU formulation restricted to the paths and cycles that are present in those sets. Then, having solved the linear programming relaxation, SSND-RAMU$(\bar{\theta}, \bar{\mathcal{P}})_{LPR}$, the approach next determines whether there are any path or cycle variables that should be added to the sets $\bar{\theta}, \bar{P}$. To determine this, we associate the dual variables ξ_h (unrestricted in sign) with each constraint (6), α_k^ψ (unrestricted in sign) with each constraint (3), $\beta_{l_t m_{t'}}^\psi$ (non-positive) with each constraint (4), and dual variables $\gamma_{l_t m_{t'}}$ with each constraint (8). With these dual variables, we estimate the reduced costs associated with cycles and paths not in the current formulation. We next describe how we use these duals in the context of generating paths and cycles.

Generating paths: With these dual variables, the formula for the reduced cost ($\bar{c}_p^{k\psi}$) associated with commodity k using path $p \in \mathcal{P}_k^\psi$ in scenario $\psi \in \Psi$ is

$$\bar{c}_p^{k\psi} = c_p^k q^{k\psi} - \alpha_k^\psi - \sum_{(l_t, m_{t'}) \in p} q^{k\psi} \beta_{l_t m_{t'}}^\psi. \tag{15}$$

Thus, after having solved SSND-RAMU$(\bar{\theta}, \bar{\mathcal{P}})_{LPR}$, the approach seeks to find paths such that $\bar{c}_p^{k\psi} < 0$, or, such that $\sum_{(l_t, m_{t'}) \in p} (c_{l_t m_{t'}}^k q^{k\psi} - q^{k\psi} \beta_{l_t m_{t'}}^\psi) < \alpha_k^\psi$. For a given commodity $k \in \mathcal{K}$ and scenario $\psi \in \Psi$, such a search can be formulated as the optimization problem of finding the shortest path from $o(k)$ to $d(k)$ in $\mathcal{G}$ with respect to the arc costs $c_{l_t m_{t'}}^k q^{k\psi} - q^{k\psi} \beta_{l_t m_{t'}}^\psi$. Such an optimization problem can be easily solved with an algorithm such as Dijkstra's (Cormen, 2009). Note the graph is acyclic as it is a time-expanded network.

Generating cycles: Given these same dual variables, we have the following formula for the reduced cost ($\bar{O}_h^\tau$) associated with having a resource that is assigned to terminal h follow the itinerary dictated by cycle $\tau \in \theta_h$,

$$\bar{O}_h^\tau = O_h^\tau - \xi_h + \sum_{(l_t, m_{t'}) \in \tau} \gamma_{l_t m_{t'}}. \tag{16}$$

As with paths, the approach seeks to find cycles such that $\bar{O}_h^\tau < 0$, or, such that $\sum_{(l_t, m_{t'}) \in \tau} \gamma_{l_t m_{t'}} < \xi_h - O_h^\tau$. Formally, given a home terminal h, this search can be formulated as the optimization problem

$$\text{minimize} \sum_{(l_t, m_{t'}) \in \tau} \gamma_{l_t m_{t'}}$$

subject to

$$\tau \in \theta_h. \tag{17}$$

In this research, θ_h is defined as the set of cycles that begin and end the planning horizon at terminal h and return to that terminal at least κ other times during the schedule length. The binary variable $v_{l_t m_{t'}}$ indicates whether arc $(l_t, m_{t'}) \in \mathcal{A}$ is in the cycle, and the binary variable η_t indicates whether the cycle should return to the home terminal in period t. Thus, to find cycles to add to $\bar{\theta}$, the approach solves the optimization problem Price-Cycle(h, γ):

$$R_h = \text{minimize} \sum_{(l_t, m_{t'}) \in \mathcal{S}} \gamma_{l_t m_{t'}} v_{l_t m_{t'}}$$

subject to

$$\sum_{t=2}^{TMAX} \eta_t \geq \kappa \tag{18}$$

$$\eta_{TMAX} = 1, \tag{19}$$

$$\sum_{(h_1, m_{t'}) \in \mathcal{A}} v_{h_1, m_{t'}} = 1, \tag{20}$$

$$\sum_{(l_{t'}, h_t) \in \mathcal{A}} v_{l_{t'}, h_t} = \eta_t, \tag{21}$$

$$\sum_{(m_{t'}, l_t) \in \mathcal{A}} v_{m_{t'}, l_t} - \sum_{(l_t, n_{t''})} v_{l_t, n_{t''}} = 0 \quad \forall l_t \in \mathcal{N}, \tag{22}$$

$$v_{l_t m_{t'}} \in \{0, 1\} \quad \forall (l_t, m_{t'}) \in \mathcal{A} \tag{23}$$

$$\eta_t \in \{0, 1\} \quad \forall t = 1, \ldots, TMAX \tag{24}$$

Constraints (18) ensure that the cycle returns to the home terminal, h, at least κ times, whereas constraints (19) ensure that the home terminal is returned to at the end of the planning horizon. Similarly, constraints (20) ensure that the cycle begins at the home terminal, h. Then, constraints (21) link the movements in the cycle to the periods when it must return home, and constraints (22) ensure that the movements can be decomposed into cycles. Finally, constraints (23) and (24) define the variables of the model and their domains. Having solved Price-Cycle(h, γ), if $R_h < \xi_h - O^{\tau}_h$, then a cycle with negative reduced cost exists which joins the set $\bar{\theta}_h$.

The complete algorithm for solving the SSND-RAMU$(\bar{\theta}, \bar{\mathcal{P}})_{LPR}$ is presented in 1. We note that the presence of paths that model direct delivery (and do not require the execution of a service) enable the algorithm to begin with each set θ_h empty. As stopping criteria, we consider a maximum number of seconds executed. We also recall that at an iteration of a column generation algorithm, a dual bound on the optimal value of the linear programming relaxation can be calculated. With this bound, the approach can determine an optimality gap. As a result the algorithm is also terminated when that optimality gap is within a pre-specified tolerance, ϵ. Finally, we let $\theta_{LPR}, \mathcal{P}_{LPR}$ denote the sets of cycles and paths at termination of Algorithm 1, and $\theta^*_{LPR} \subseteq \theta_{LPR}, \mathcal{P}^*_{LPR} \subseteq \mathcal{P}_{LPR}$ the sets of paths and cycles used in the final primal solution to SSND-RAMU$(\bar{\theta}_l, \bar{\mathcal{P}}_k)_{LPR}$ produced by Algorithm 1.

Algorithm 1 Solve-SSND-RAMU$_{LPR}$

```
Initialize $\bar{\mathcal{P}}^{\psi}_k$ with path that models direct delivery, $\forall k \in \mathcal{K}, \psi \in \Psi$
Set $\bar{\theta}_h = \emptyset, \forall h \in \Lambda$
Set $\bar{\theta} = \bigcup_{h \in \Lambda} \bar{\theta}_h$ and $\bar{\mathcal{P}} = \bigcup_{k \in \mathcal{K}, \psi \in \Psi} \bar{\mathcal{P}}^{\psi}_k$
while stopping criteria not met do
  Solve SSND-RAMU$(\bar{\theta}, \bar{\mathcal{P}})_{LPR}$ for dual variables $\xi_h, \alpha^{\psi}_k, \beta^{\psi}_{l_t m_{t'}}, \gamma_{l_t m_{t'}}$
  for all $k \in \mathcal{K}, \psi \in \Psi$ do
    Find shortest path from $o(k)$ to $d(k)$ with respect to arc costs $c^k_{l_t m_{t'}} - q^{k\psi} \beta^{\psi}_{l_t m_{t'}}$
    If shortest path distance $< \alpha^{\psi}_k$ then add path to $\bar{\mathcal{P}}^{\psi}_k$.
  end for
  $\bar{\mathcal{P}} = \bar{\mathcal{P}} \bigcup_{k \in \mathcal{K}, \psi \in \Psi} \bar{\mathcal{P}}^{\psi}_k$
  for all $h \in \Lambda$ do
    Solve Price-Cycle$(h, \gamma)$ for value $R_h$ and cycle $\tau$
    if $R_h < \xi_h - O^{\tau}_h$ then
      Add $\tau$ to $\bar{\theta}_h$
    end if
  end for
  $\bar{\theta} = \bar{\theta} \bigcup_{h \in \Lambda} \bar{\theta}_h$
end while
```

5.2 The CG-based heuristic

The first heuristic presented begins by solving SSND-RAMU$_{LPR}$ to generate the sets $\theta_{LPR}, \mathcal{P}_{LPR}$. These sets of cycles and paths are then used to formulate and solve SSND-RAMU$(\theta_{LPR}, \mathcal{P}_{LPR})$ with a commercial mixed integer programming solver. Formally, Algorithm 2 presents this approach, which we call *CG-Solve*.

Algorithm 2 CG-based heuristic (CG-Solve)

```
Solve SSND-RAMU$_{LPR}$ with Algorithm 1
Choose all paths ($\mathcal{P}_{LPR}$) and cycles ($\theta_{LPR}$) generated
Solve SSND-RAMU($\theta_{LPR}, \mathcal{P}_{LPR}$) with a MIP solver
```

5.3 IP-Solve

The second proposed algorithm for solving the SSND-RAMU$(\bar{\theta}, \bar{\mathcal{P}})$ is called *IP-Solve* and is a matheuristic wherein a neighborhood of a solution is defined and searched through the formulation and solution of a mixed integer program. At an iteration of this matheuristic, we presume a known solution *sol* with $\mathcal{P}_{sol}$ and θ_{sol} representing the sets of paths and cycles used in that solution. Next, we determine the neighborhood to search, wherein a neighborhood includes both a set of paths that can be taken in each scenario, $\mathcal{P}_{nbhd} = \cup_{\psi \in \Psi} \mathcal{P}^{\psi}_{nbhd}$ chosen from those known, $\mathcal{P}_{cand}$, and a set of cycles, θ_{nbhd}, chosen from those known. Then, to search that neighborhood we solve SSND-RAMU$(\theta_{sol} \cup \theta_{nbhd}, \mathcal{P}_{sol} \cup \mathcal{P}_{nbhd})$ with an off-the-shelf optimization solver. Note that in the next discussion an overline (e.g. $\bar{z}^{\tau}$) indicates the value of a variable in the solution *sol*. A formal description of the methodology is presented in Algorithm 3.

Algorithm 3 IP-based Mathheuristic (IP-Solve)

```
Solve SSND-RAMU$_{LPR}$ via column generation for cycles $\theta^*_{LPR}$ and paths $\mathcal{P}_{LPR}$
Solve SSND-RAMU($\theta^*_{LPR}, \mathcal{P}_{LPR}$) to produce solution $sol = (\bar{z}, \bar{y}, \bar{x}), \theta_{sol}, \mathcal{P}_{sol}$
Set $\theta_{cand} = \theta_{LPR}$, $\mathcal{P}_{cand} = \mathcal{P}_{LPR}$
while stopping criteria not met do
  Determine neighborhood to search
  if searching neighborhood involves generating new cycles and paths then
    Generate cycles, $\theta_{new} \in \theta \setminus \theta_{cand}$, and paths, $\mathcal{P}_{new} \in \mathcal{P} \setminus \mathcal{P}_{cand}$
    Set $\theta_{cand} = \theta_{cand} \cup \theta_{new}$, $\mathcal{P}_{cand} = \mathcal{P}_{cand} \cup \mathcal{P}_{new}$.
  end if
  Determine $\theta_{nbhd} \in \theta_{cand} \setminus \theta_{sol}$ and $\mathcal{P}_{nbhd} \in \mathcal{P}_{cand} \setminus \mathcal{P}_{sol}$ based on neighborhood
  Solve SSND-RAMU($\theta_{sol} \cup \theta_{nbhd}, \mathcal{P}_{sol} \cup \mathcal{P}_{nbhd}$) for solution $sol, \theta_{sol}, \mathcal{P}_{sol}$
end while
```

We consider searching two different neighborhoods in the course of executing IP-Solve. To create the first neighborhood, called *CG-Nbhd*, we first generate new cycles and paths before determining which cycles to include in θ_{cand}. However, the neighborhood consists of all known paths (e.g., we set $\mathcal{P}_{nbhd} = \mathcal{P}_{cand} \setminus \mathcal{P}_{sol}$). While the first neighborhood consists of a subset of known cycles, but all known paths, the second neighborhood, called *ScenPath-Nbhd*, does the opposite. To construct this neighborhood, we include all known cycles, but only the paths for a limited set of scenarios. We next describe these two neighborhoods in detail.

5.4 *CG-Nbhd*

To create this neighborhood, we first generate new cycles and paths, and then determine which cycles to include in θ_{nbhd}. As noted, the neighborhood consists of all known paths (e.g., $\mathcal{P}_{nbhd} = \mathcal{P}_{cand} \setminus \mathcal{P}_{sol}$). As such, we first describe how new cycles and paths are generated, and then how we determine which cycles to include in θ_{nbhd}.

To generate new cycles and paths a restricted instance of the SSND-RAMU$(\theta_{cand}, \mathcal{P}_{cand})_{LPR}$ is solved with column generation. To restrict the instance, the approach partitions the cycles in θ_{cand} into two sets: (1) θ^{one}_{cand}, which contains cycles that must be selected, and, (2) θ^{zero}_{cand}, which contains cycles that can not be selected. Then, with these sets, a partial solution to SSND-RAMU$(\theta_{cand}, \mathcal{P}_{cand})_{LPR}$ can be created by fixing the value of the variable z^{τ}_{h} to $1(0)$ when $\tau \in \theta^{one}_{cand}$ $(\tau \in \theta^{zero}_{cand})$. Having fixed these variables, 1 then solves this restricted instance. Note that to ensure that a cycle fixed to zero (e.g., $\tau \in \theta^{zero}_{cand}$) is not generated by the column generation procedure, we modify the pricing problem, Price-Cycle(h), to include a cardinality constraint $\sum_{(l_t,m_{t'})\in\tau^{zero}} a_{l_t m_{t'}} \leq (\sum_{(l_t,m_{t'})\in\tau^{zero}} 1) - 1$. We add such a constraint to Price-Cycle(h) for each cycle with origin h that is fixed to zero.

The utilization of each cycle in the current solution informs the sets θ^{one}_{cand} and θ^{zero}_{cand}. Specifically, each cycle $\tau \in \theta_{sol} \cap \theta_{cand}$ has a score assigned,

$$\sigma_{\tau} = \sum_{\psi\in\Psi} \sum_{(l_t,m_{t'})\in\tau} \phi_{\psi}\Big(\sum_{k\in\mathcal{K}} \sum_{p\in\bar{\mathcal{P}}_k:(l_t,m_{t'})\in p} q^{k\psi}\bar{x}^{k\psi}_{p}\Big)/u_{l_t m_{t'}},$$

that measures the expected utilization of services in that cycle in the current solution. We then sort the cycles in θ_{sol} in descending order of σ_{τ}, and put the first F (an algorithm parameter) into the set θ^{one}_{cand} and the remaining in θ^{zero}_{cand}. As a result, when solving the restricted instance of SSND-RAMU$(\theta_{cand}, \mathcal{P}_{cand})_{LPR}$, Algorithm 1 will generate new cycles to complement those that are most utilized in the current solution.

Regarding the cycles to include in θ_{nbhd}, by solving SSND-RAMU$(\theta_{cand}, \mathcal{P}_{cand})_{LPR}$, an examination of the z^{*}_{LPR} values determines which of the new cycles best complement those included in θ^{one}_{cand}. Specifically, Algorithm 1 generates new cycles, labeled θ_{new}, sorted in descending order of the value z^*_{LPR}. The set θ_{nbhd} includes the first C (an algorithm parameter) of those. New paths, $\mathcal{P}_{new}$, are added to $\mathcal{P}_{cand}$ and thus included in $\mathcal{P}_{nbhd}$. Figure 2 provides a high-level flow chart of how this neighborhood is constructed.

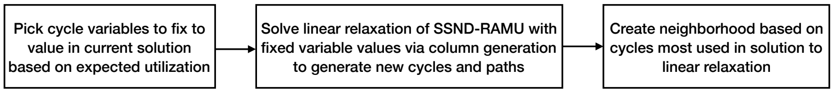
Pick cycle variables to fix to value in current solution based on expected utilization
Solve linear relaxation of SSND-RAMU with fixed variable values via column generation to generate new cycles and paths
Create neighborhood based on cycles most used in solution to linear relaxation

Figure 2: *CG-Nbhd*: A neighborhood based on generating cycles and paths

5.5 *ScenPath-Nbhd*

Whereas the previous neighborhood consists of all known paths and a limited set of cycles, this neighborhood consists of all known cycles (e.g., $\theta_{nbhd} = \theta_{cand} \setminus \theta_{sol}$) and a limited set of paths. In particular, in this neighborhood, for some scenarios a commodity may only take a path that it takes in the current solution, whereas for others, any known paths may be taken.

To instantiate this neighborhood, the approach selects a subset of scenarios, $\bar{\Psi} \subseteq \Psi$, wherein $|\bar{\Psi}|$ is an algorithm parameter. Then, for $\psi \in \Psi \setminus \bar{\Psi}$, $\mathcal{P}^{\psi}_{nbhd}$ includes paths p wherein $\bar{x}^{kw}_p > 0$. Thus, the scenarios in $\Psi \setminus \bar{\Psi}$ are the ones where the paths a commodity may take are limited to those currently taken. Alternately, for $\psi \in \bar{\Psi}$, the approach sets $\mathcal{P}^{\psi}_{nbhd} = \mathcal{P}^{\psi}_{cand}$. In other words, scenarios in $\bar{\Psi}$ are the ones where a commodity can follow any known path. The set $\bar{\Psi}$ is determined randomly.

6 Experimental Design

In this section, we describe the experiments used to validate the model, SSND-RAMU, and assess the performance of the proposed solution approaches. We first discuss the transportation network that all instances are based on as well as the geographic and temporal aspects of commodities. We then discuss the distributions used for freight volumes and how scenarios were created to model those distributions. We next provide a discussion of the values used for model parameters and how they were derived. We finish with a discussion of the algorithm parameter values and the computational setting used for experiments.

6.1 Network and shipments

As this research is somewhat inspired by the planning operations of a Less-than-truckload freight transportation carrier, we derive the instances used in the computational study from a network that mimics the hub-and-spoke structure often seen in LTL networks. Specifically, all instances are based on the network illustrated in Figure 3. Regarding transportation time, and recalling that a time-space network models time in periods, we presume that all moves within a region require one period of time, whereas inter-regional moves require two periods of time. With this network we model two layers of hubs, with the first (nodes H1, H2, H5, and H6) serving as consolidation points for satellites and the second (nodes H3 and H4) serving as consolidation points for their respective regions (although shipments need not be transferred at those terminals to depart/enter a region). Regarding the time-space network, we model a 6-day week, with two periods per day. Thus, the time-space network on which we plan has 144 nodes ($|\mathcal{N}| = 144$) and 600 services ($|\mathcal{S}| = 600$).

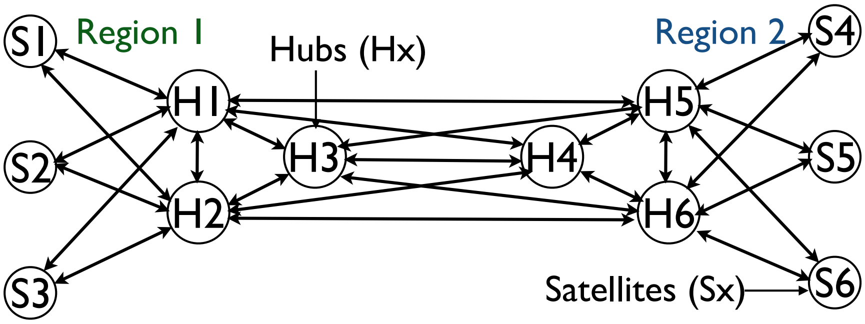

Figure 3: Hub and spoke network used in experiments

We also presume the same set of shipments in each instance, with respect to origin and destination terminal in the network given in Figure 3 (although as we will discuss later we allow shipment volumes to vary). Regarding shipments, Table 3 describes their distribution across origins and destinations in the network. Here, for each pair of nodes of a given type (e.g. (Sx, Sx)), "Number" refers to the number of shipments for that type. Then, "Frequency" refers to how often each shipment appears during a week. In other words, the first line of the table indicates that there are three shipments of the form (S1,Sx) (e.g., (S1, S2), (S1, S4), (S1,S3)), each shipment occurring only once a week. The last line indicates that there are five shipments of the form (H1, Hx) and each appears three times during the week. Note that we do not presume the same volumes for different shipments with the same origin and destination. In other words, a (H1,H6) shipment that originates in period two can have a different volume than the (H1,H6) shipment that originates in period six. However, the service standard (the number of periods it can take to deliver a shipment from its origin to its destination) is determined solely by the origin and destination terminals. In sum, the instances consist of 228 commodities.

Origin	Destination	Number	Frequency
Sx	Sx	3	1
Sx	Hx	5	2
Hx	Sx	5	2
Hx	Hx	5	3

Table 3: Shipments in instance

6.2 Freight volume distributions and scenarios

Regarding freight volumes, using one week's worth of volume data from a US Less-than-truckload carrier we derived that volumes could be approximated with a General Beta distribution of the form $d = 150 + 21,050Z$ wherein Z followed a beta distribution with parameters $a = 0.057706$ and $b = 7.79$. In addition, our analysis of shipment volume correlations in the data suggested there were none that were statistically significant.

To ensure that we assess the effectiveness of the proposed algorithms on a set of scenarios that accurately represents these distributions we used the algorithm described in Høyland et al. (2003) to generate sets of scenarios. Note that this algorithm takes as input the number of scenarios to generate and reports (along with a set of scenarios) how well the moments of the underlying distribution are matched. It also takes a correlation matrix for the random variables. As the size of an instance of the SSND-RAMU grows in the number of scenarios, the effectiveness of the proposed algorithms is also dependent on the number of scenarios. Thus, by using this algorithm we were able to derive a set of scenarios that both closely approximates the moments of shipment volume distributions and yields instances of the SSND-RAMU that are manageable for the proposed algorithms.

Specifically, when generating scenarios, we presumed that the volume of all 228 commodities followed the General Beta distribution described above, and that their volumes were independent. Then, given these distributions, we determined the four moments of that distribution. Based on our analysis of correlations in the data, we provided the Identity matrix as a correlation matrix to the scenario generation algorithm. We found that the smallest number of scenarios that yielded a close approximation of the first four moments as well as the given correlation matrix was 24.

We also modify the beta distribution described above to model higher volumes and greater variance in volumes. Specifically, we generalize the formula $d = 150 + 21,050 * Z$ for determining freight volumes to $d = 150 + 21,050 * v_d * Z$ and consider the values 1 and 5 for v_d. We also consider two more distributions for freight, with the first having twice the standard deviation of the fitted distribution ($m_\sigma = 2$) and the second having three times the standard deviation ($m_\sigma = 3$).

6.3 Model parameter values

Regarding parameter values, we consider the cost of outsourcing a service (both via contract and on the spot market) to be a multiple of the underlying service cost. Specifically, the parameter μ^e sets $f^e_{lm} = \mu^e f_{lm}$ and the parameter μ^σ sets $f^\sigma_{lm} = \mu^\sigma f_{lm}$. The following combinations of values for these parameters create the instances: $(\mu^e, \mu^\sigma) = (2, 3); (2, 4); (3, 4)$. Note that in the experiments, we focus on acquiring new resources only. Thus, assume $I_w = 0, \forall w \in \Lambda$. As such, we need not consider the parameter value $F_{wh}, w \in \Lambda$. Table 4 summarizes the different instance parameter values used.

Cost parameter	Values considered	Distribution parameter	Values considered
μ^e	2,3	v_d	1,5
μ^σ	3,4 (note we never consider $\mu^e = \mu^\sigma$)	m_σ	1,2,3
	Cost structure	1,2,3,4	

Table 4: Instance parameter values

We next turn to the cost associated with acquiring a resource for a terminal (parameter F_{Ah}). We consider four cost structures, detailed in Table 5. While the first three structures are used to model different practical settings, the last is used to validate that the solution approach is producing sensible solutions.

Structure	Resource acquisition and allocation costs
1	\$1,800 at each terminal
2	\$1,800 at satellites, \$2,000 at hubs
3	\$2,000 at satellites, \$900 at hubs
4	\$1,800 at S1,H1,H3,S6; \$1,900 at S2,H6,S4; \$2,000 all other terminals

Table 5: Cost structures for acquiring resources

In summary, we consider 72 different instances, all of which are defined on a time-space network with 144 nodes and 600 arcs. They each have 228 commodities, whose volumes are modeled with 24 scenarios.

6.4 Computational setting

In all experiments, we executed both CG-Solve and IP-Solve on each of the 72 instances on a cluster of machines with 8 Intel Xeon CPUs running at 2.66 GHz with 32 GB RAM. All linear and mixed integer programs were solved with CPLEX 12. When executing CG-Solve, we solved the MIP, SSND-RAMU($\theta_{LPR}, \mathcal{P}_{LPR}$), with a time limit of five hours and optimality tolerance of 1%. We let IP-Solve execute for 90 minutes and during its execution, all MIPs were solved with a time limit of 60 seconds and optimality tolerance of 1%.

The time spent solving SSND-RAMU($\bar{\theta}, \bar{\mathcal{P}})_{LPR}$ at the beginning of CG-Solve and IP-Solve was limited to 10 minutes. When solving SSND-RAMU($\bar{\theta}, \bar{\mathcal{P}})_{LPR}$ to generate neighborhood *CG-Nbhd* in the context of IP-Solve, Algorithm 1 performed for one iteration. Fundamentally, IP-Solve requires values for three parameters: F, C, and $|\bar{\Psi}|$. We used the same values for these parameters in all experiments and present those values in Table 6.

F	C	$\lvert\bar{\Psi}\rvert$
5	40	5

Table 6: Algorithm parameter values

7 Value of modeling uncertainty in SSND-RAMU

In this section, we assess the value of explicitly recognizing uncertainty in the SSND-RAMU on the instances described in Section 6. To do so, we calculate the *Value of the Stochastic Solution* (VSS). As fast solution times are not our objective in this section, we execute CG-Solve, albeit with the MIP solved at the end given a time limit of ten hours. Specifically, to calculate the VSS, we first determine the mean-value scenario, $\bar{\psi}$, wherein the volume of each commodity is set to its mean, $\bar{q} = \sum_{\psi \in \Psi} \phi_w q^{k\psi}$. Then, CG-Solve is used to solve the SSND-RAMU formulated with just the mean-value scenario $\bar{\Psi}$. This yields decisions regarding resource acquisition ($\bar{a}_{wh}$), resource routes ($\bar{z}^{\tau}_{h}$), services operated by owned resources ($\bar{y}_{l_t, m_{t'}}$), and services operated by third-party resources ($\bar{y}^{e}_{l_t, m_{t'}}$). These decisions are then evaluated in the second stage, and for each scenario, by solving SSND-RAMU($\theta_{LPR}, \mathcal{P}_{LPR}$) with the first-stage variables fixed to those values. Doing so yields a total cost, labeled obj_{mean}, that consists of the first-stage costs associated with those decisions and the resulting expected costs in the second stage.

Next, for that same instance albeit with all scenarios, we again execute CG-Solve, and again with the MIP SSND-RAMU($\theta_{LPR}, \mathcal{P}_{LPR}$) solved for ten hours, to derive the total cost $obj_{SSND-RAMU}$. The VSS is calculated as the gap between these two costs, $(obj_{mean} - obj_{SSND-RAMU})/obj_{mean}$. Table 7 reports averages of these gaps, by variation (m_{σ}) and cost structure. We note that CPLEX was able to solve each instance of the mean-value problem to a 1% optimality tolerance, while the resulting second stage problems were also solved to within 1%. CPLEX could always solve the SSND-RAMU($\theta_{LPR}, \mathcal{P}_{LPR}$) to within 5% of optimality. As such, the gaps reported in Table 7 somewhat under-estimate the VSS when the SSND-RAMU is restricted to the cycles θ_{LPR} and paths $\mathcal{P}_{LPR}$.

Not surprisingly, the largest levels of variation ($\mu_{\sigma} = 3$) lead to the largest gaps.

Table 7: Value of stochastic solution

	Cost structure				
μ_σ	1	2	3	4	Average
1	6.40%	7.66%	1.06%	6.87%	5.50%
2	5.07%	4.83%	2.92%	7.08%	4.98%
3	7.50%	8.53%	2.98%	7.24%	6.56%
Average	6.32%	7.01%	2.32%	7.06%	5.68%

Interestingly, cost structure 3 results in the smallest gaps. We hypothesize that the cost structure so favors the allocation of acquired resources to hubs that explicitly modeling uncertainty (as done with SSND-RAMU) leads to the same first-stage decisions as not doing so. We conclude from these results that there is value in solving the SSND-RAMU instead of solving the mean-value scenario problem. We will report on a similar analysis later in the paper when we discuss the application of the algorithm to a instances derived from the operations of a European postal carrier.

8 Assessing effectiveness of IP-Solve

In this section, we assess the ability of the proposed solution approach IP-Solve to produce high-quality solutions by benchmarking its performance against that of CG-Solve. To do so, we execute both algorithms (CG-Solve and IP-Solve) on each instance, yielding two objective function values for each instance: (1) $obj_{IP-Solve}$, the objective function value of the best solution found by IP-Solve, and, (2) $obj_{CG-Solve}$, the objective function value of the best solution found by CG-Solve. We also consider a third objective function value, $obj_{CG-Solve}^{IP-Solve-TTB}$, which is the best solution found by CG-Solve by the time IP-Solve found its best solution. The analysis of the performance of IP-Solve is based on the following gaps:

$$gap_{CG-Solve}^{IP-Solve} = (obj_{CG-Solve} - obj_{IP-Solve})/obj_{CG-Solve},$$
$$gap_{CG-Solve}^{IP-Solve-TTB} = (obj_{CG-Solve}^{IP-Solve-TTB} - obj_{IP-Solve})/obj_{CG-Solve}^{IP-Solve-TTB}.$$

We present a distribution of these gaps in Figure 4 (e.g., the percentage of instances for each % wherein the gap is in that range). We note that for every instance $gap_{CG-Solve}^{IP-Solve}$ is positive, meaning IP-Solve produced a better solution than CG-Solve. In addition, the average of $gap_{CG-Solve}^{IP-Solve}$ over all instances is 5.44% and the average of $gap_{CG-Solve}^{IP-Solve-TTB}$ is 5.62%. While the gaps between two and three percent are the most frequent, nearly 60% the instances have a gap greater than 3%. Recalling that IP-Solve executes for a little over an hour while CG-Solve executes for five hours leads to the conclusion that IP-Solve is superior to CG-Solve.

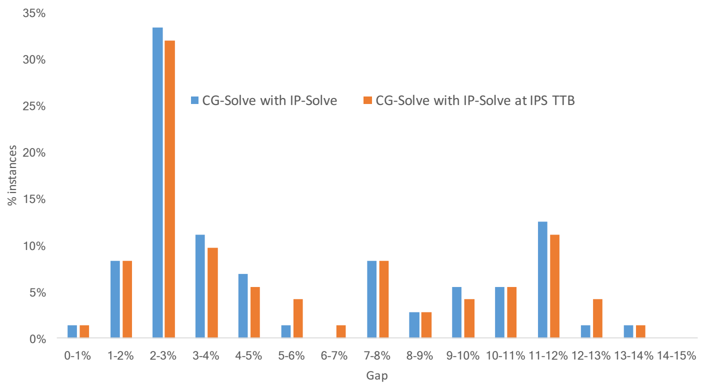

Figure 4: IP-Solve vs CG-Solve

We next turn our attention to the length of time required by IP-Solve to find its best solution and how much better that best solution is than the first one found by IP-Solve. Averaging over all instances, it takes IP-Solve 3,339.43 seconds (roughly 56 minutes) to find its best solution. The gap between the first and best solutions found is calculated as $gap^{best}_{first} = (obj_{first} - obj_{best})/obj_{first}$, wherein obj_{first} is the objective function value of the initial solution found and obj_{best} is the objective function value of the best solution found. On average, $gap^{best}_{first} = 7.17\%$.

We next study how the two different neighborhoods contribute to the search for improving solutions. For each instance, we calculate what proportion of gap^{best}_{first} can be attributed to searching each neighborhood. Similarly, for each instance we calculate the percentage of time when searching each neighborhood produced an improving solution. Table 8 reports averages of these two statistics over all instances and for each neighborhood. While both neighborhoods yield improving solutions, searching *CG-Nbhd* accounts for the majority of the improvement and often yields an improving solution.

Table 8: Neighborhood statistics

Type	% total improvement	% times found improving
CG-Nbhd	81.14%	72.15%
ScenPath-Nbhd	18.86%	64.69%

We next assess whether the performance of either approach is sensitive to the instance parameters reported in Table 4. To measure this, we average $gap^{IP-Solve}_{CG-Solve}$ over all instances with the same parameter value (e.g., all instances wherein $\mu^e = 3$). Table 9 presents results according to cost-based parameters, while Table 10 presents results according to demand-based parameters. The algorithm is relatively robust with respect to all instance parameters; the only parameter that impacts $gap^{IP-Solve}_{CG-Solve}$ is the demand multiplier, v_d. We hypothesize that a smaller demand multiplier leads to *CG-Solve* solving an integer program with a weak linear programming relaxation, and thus makes it harder for it to find a high-quality solution.

Table 9: Algorithm performance by cost-based instance parameters

Parameter	Contract outsourcing μ^e		Spot outsourcing μ^σ		Cost structure			
Value	2	3	3	4	1	2	3	4
$gap_{CG-Solve}^{IP-Solve}$	5.22%	5.87%	5.72%	5.29%	6.20%	6.58%	2.58%	6.38%

Table 10: Algorithm performance by freight demand-based instance parameters

Parameter	Volume v_d		Variation m_σ		
Value	1	5	1	2	3
$gap_{CG-Solve}^{IP-Solve}$	8.43%	2.45%	5.24%	5.21%	5.86%

To produce a high-quality solution, IP-Solve must both produce the cycles and paths that are needed for a good solution and construct a good solution out of those cycles and paths. With this last set of experiments we test the ability of IP-Solve to produce a high quality solution given a set of paths and cycles. To that effect, and to further understand the quality of the solution produced by IP-Solve, we ran one more set of experiments wherein after IP-Solve terminated, we solve SSND-RAMU(θ_{cand}, $\mathcal{P}_{cand}$) as a MIP with CPLEX for five hours to generate a primal solution with objective function value, $obj_{CPLEX-5hours}$, and dual bound, $bound_{CPLEX-5hours}$. Note that CPLEX was seeded with the best solution found by IP-Solve as a starting solution. We then calculate the gap in the objective function value between the best solution produced by IP-Solve and the best solution produced by CPLEX after solving SSND-RAMU(θ_{cand}, $\mathcal{P}_{cand}$), calculated as $primal_{IP-Solve}^{CPLEX-5h} = (obj_{IP-Solve} - obj_{CPLEX-5hours})/obj_{IP-Solve}$. We also calculate the optimality gap associated with the solution produced by IP-Solve and the dual bound found by CPLEX, $opt_{IP-Solve}^{CPLEX-5h} = (obj_{IP-Solve} - bound_{CPLEX-5hours})/obj_{IP-Solve}$. Table 11 reports averages of these gaps over all instances with the same volume and variation multiplier.

Table 11: Comparison with CPLEX solving SSND-RAMU($\theta_{cand}, \mathcal{P}_{cand}$) for five hours

	$primal_{IP-Solve}^{CPLEX-5h}$				$opt_{IP-Solve}^{CPLEX-5h}$			
	Cost structure				Cost structure			
v_d	1	2	3	4	1	2	3	4
1	2.14%	2.17%	3.41%	2.26%	25.66%	26.24%	20.03%	26.48%
5	2.23%	1.84%	1.93%	1.99%	5.91%	5.43%	4.53%	5.69%

The $primal_{IP-Solve}^{CPLEX-5h}$ results indicate that even with five hours of solving time, CPLEX was unable to produce a solution of significantly higher quality than that produced by IP-Solve. Considering the $opt_{IP-Solve}^{CPLEX-5h}$ results when volumes are high, IP-Solve is producing solutions that are within 6% of optimal, given the set of cycles θ_{cand} and paths

$\mathcal{P}_{cand}$. We believe the large optimality gaps when volumes are low can be attributed to a weak linear programming relaxation for those instances. These results are further evidence of the ability of IP-Solve to produce high-quality solutions. Thus, we next turn our attention to assessing the potential of the model and IP-solve for assisting with the planning of a real-world operation.

9 Application to Operations of a European Postal Carrier

In this section, we illustrate the ability of the model and solution method to assist in planning the operations of a European postal carrier. This postal carrier provides same and next-day ground-based transportation services for parcels and small packages. To do so, it employs a network of over 200 consolidation terminals; however, the overwhelming majority of drivers serving this network are domiciled at 14 of the primary terminals, which we will focus on for this analysis. The hourly wages of these drivers vary based on their domicile. We illustrate this variation in Figure 5, which depicts the driver hourly wage rate for each terminal as a percentage of the lowest wage rate across all terminals.

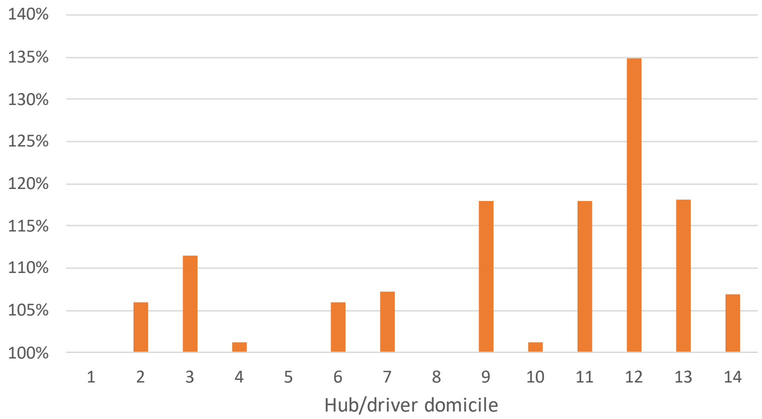

Figure 5: Driver hourly wages for each terminal, relative to minimum

Given this variation in hourly wages, the postal carrier sought to determine the number of drivers domiciled at each terminal in a way that reduced labor costs and ensured the timely delivery of packages and parcels (which we will now refer to as "shipments"). To serve that goal, we applied the model and solution approach, IP-Solve, proposed in this paper to indicate where drivers should be domiciled as well as provide a two-day repeatable schedule of transportation services and driver schedules on which the carrier could base their operations. We will next discuss this application, albeit with an emphasis on the ability of IP-Solve to effectively solve instances associated with this real-world application.

We first discuss the derivation of the instance of the SSND-RAMU used to assist the postal carrier's planning. While the carrier routed shipments through a network of over 200 terminals, we derived an instance of the SSND-RAMU based on 14 of the primary terminals. Based on proximity to large metropolitan areas and position in the network, these terminals are the most likely domiciles for drivers. Also, over 80% of the package distances traveled occur between those 14 terminals. Regarding services, the carrier executed 182 transportation moves between these terminals. Thus, the physical network used to build instances of the SSND-RAMU for this application had $|\Lambda| = 14, |\mathcal{S}| = 182$. To create the time-space network, $\mathcal{G}$, we modeled 12 periods per day ($TMAX = 24$). Thus, the time-space network considered in this real-world application consisted of 168 nodes and 4,367 arcs.

Given the goals of the postal carrier, in this application resources correspond to drivers. As the carrier sought to develop "targets" for the numbers of drivers domiciled at terminals, we modeled each terminal as beginning with 0 drivers ($I_w = 0, \forall w \in \Lambda.$). As we sought to develop a two-day repeatable schedule of operations, we determined the cost, F_{wh}, associated with allocating a driver to terminal w based on the total hourly wages paid over a two-day period given the hourly wage rate associated with terminal w. We note that in this case the costs associated with allocating drivers to terminals is on the same scale as the transportation costs incurred over the two-day schedule. Cycles were constructed such that European hours-of-service guidelines were upheld and the driver returned to his/her domicile for a rest period at least once over the two-day schedule.

Transportation costs, which were represented in the service fixed costs, f_{lm}, were determined based on per-kilometer costs reported by the carrier. The cost associated with outsourcing a service on a contract basis, f^e_{lm}, was based on a per-kilometer transportation cost that differed by the terminal, l, from which the service executed. As this carrier did not use a spot market to acquire capacity, variables $y^{\sigma\psi}_{l_t m_{t'}}$ were set to 0 in all instances. Finally, path costs, c^k_p, reflected holding costs associated with loading and unloading shipments at terminals (with these per-shipment costs also differing by terminal).

Regarding shipments, there were 316 unique combinations of (origin terminal, destination terminal, available period) over an average two day segment, each of which we modeled as a commodity $k \in \mathcal{K}$. We derived distributions for shipment volumes from one month of historical data, with an average of approximately 360,000 new packages requiring service each day. One month provided 20 days of data as we did not include weekends, when demand was considerably lower. We again used the algorithm described in Høyland et al. (2003) to generate scenarios, although for this data we needed 50 scenarios to reasonably approximate the four moments of the distributions. An analysis of the data suggested there was little correlation between the volumes associated with different (origin terminal, destination terminal, available period) combinations. To test the effectiveness of IP-Solve under different shipment transit times from origin to destination terminals, we created three instances that differed in the number of periods each shipment had to reach its destination terminal. The first instance was based on a baseline transit time derived from discussions with the carrier. In the second (third) instance, each shipment was allowed its baseline plus one (two) periods to travel from its origin to its destination. As discussed previously, we associate with each shipment one path that models direct delivery from the shipments' origin to its destination by an external carrier.

We executed IP-Solve on a version of the SSND-RAMU customized to the operations of this carrier (e.g. constraints (7) were not included) with this data. Similarly, the pricing problem, Price-Cycle(h, γ) was customized to better approximate the rules this carrier seeks to observe when planning driver schedules. We do not discuss these customizations in detail, both for the sake of brevity and because they do not impact the overall design and execution of IP-Solve.

As with the other set of instances, we benchmarked the performance of IP-Solve against that of CPLEX solving an instance of the $SSND - RAMU(\theta_{cand}, \mathcal{P}_{cand})$ formulated with the cycles and paths found by IP-Solve and seeded with a starting solution corresponding to the best solution found by IP-Solve. We report in Table 12 the same statistics as those reported in Table 11 and see that in five additional hours, CPLEX was able to improve on the best solution found by IP-Solve by 5.84%, on average. We also see that the greater the flexibility in operations (i.e. longer shipment transit times), the better IP-Solve performs. We conclude from these results that IP-Solve can be effective at producing high-quality solutions on instances derived from real applications.

Table 12: Comparison of IP-Solve with CPLEX solving SSND-RAMU$(\theta_{cand}, \mathcal{P}_{cand})$ for five hours on real-world instances

$primal^{CPLEX-5h}_{IP-Solve}$			$opt^{CPLEX-5h}_{IP-Solve}$		
Shipment transit time			Shipment transit time		
1	2	3	1	2	3
7.60%	5.45%	4.45%	11.07%	9.47%	9.00%

As the carrier did not acquire capacity from a spot market, these instances provide another method for assessing the value of solving a model that explicitly recognizes uncertainty. Specifically, as the solution to the mean-value scenario problem may not establish sufficient capacity to transport shipments in all scenarios, shipments may have to be routed on an externally-supported direct delivery path in some scenarios. As we do not have data regarding how much such a direct delivery path would actually cost, we present in Table 13 the percentage of such shipments, averaged over all scenarios.

We note that in the solutions to the SSND-RAMU, no direct delivery paths were used. While these percentages of paths are not large, they give another indicator that explicitly modeling uncertainty leads to a more robust allocation of capacity.

Table 13: % Shipments must be transported on direct delivery path, averaged over all scenarios

% Shipments		
Shipment transit time		
1	2	3
2.20%	2.20%	1.88%

10 Conclusions and Future Work

We focused in this paper on modeling and solving in a unified way two planning problem faced by consolidation-based freight transportation carriers: selecting and scheduling the set of services required to route shipments while meeting the economic goals of the company and the service standards customers expect, and, selecting and efficiently routing the resources required to provide this service, while observing governmental (and other) regulations. We proposed SSND-RAMU, a scheduled service network design model that simultaneously addresses strategic decisions on fleet sizing and allocation, including acquisition and outsourcing, and tactical decisions on building the transportation plan and schedule. Moreover, as a well-sized fleet and a well-designed transportation plan should be able to accommodate fluctuations in freight volumes, we explicitly addressed uncertainty in demand (freight volumes) through the use of scenarios, which makes SSND-RAMU a stochastic program. Solving this program will assist transportation companies to size, locate, and use their fleet while recognizing that customer demands for transportation services are not known with certainty.

Given the computational difficulties associated with solving stochastic programs exactly, we proposed two column-generation-based matheuristics for addressing the model. The matheuristics framework we propose decomposes the optimization problem across multiple dimensions, and evaluates a neighboring solution across all scenarios. This is the first heuristic scheme for this class of problems. Extensive computational experiments on two sets of instances show that it is effective; the second matheuristic, called IP-Solve, being superior. The second set of instances was derived from the operations of a European postal carrier, with results on those instances suggesting that IP-Solve can benefit practice.

While the proposed matheuristic is able to produce high-quality solutions relative to a benchmark method, it is often unable to produce strong enough lower bounds to derive meaningful optimality gaps. While many polyhedral analyses of service network design problems have yielded valid inequalities that greatly strengthened those formulations, the model considered in this paper considers a much richer set of decisions. As a result, one avenue of future work is to perform a polyhedral analysis of this formulation, with an eye towards generating valid inequalities that strengthen its linear programming relaxation. Clearly, another avenue of future research is to embed IP-Solve within a branch-and-bound-type search framework, yielding an exact method.

The model assumes that resources remain at the terminal to which they are allocated. However, it is not uncommon for a resource to be allocated to one terminal during one season and then re-allocated to another terminal in the next season. As such, whereas our model implicitly considers a single season, we are exploring extending it to instead consider multiple seasons. This new model will capture that resources are acquired and allocated, and then can be repositioned at the beginning of each of the subsequent seasons. Such a model will likely be a multi-stage stochastic program and will necessitate new algorithmic developments.

Similarly, we are exploring other heuristic strategies for stochastic programs. The matheuristic presented calculates the resource cost explicitly for each neighboring solution (through the solution of a two stage stochastic mixed integer program). However, researchers have had great success in other problem settings with algorithms that approximate the recourse cost, either with linear inequalities, or by explicitly recognizing a modified (likely smaller) set of scenarios. As such we are developing a matheuristic for two stage stochastic programs that, at each iteration, solves a stochastic program that approximates the recourse cost.

Relating to practice, the effectiveness of the proposed heuristics opens the door to a detailed empirical study of how operational settings impact managerial decision-making, with respect to both the strategic and tactical levels. We anticipate performing such a study, in the context of the European postal carrier discussed in this paper.

Acknowledgments

Partial funding was provided by the Natural Sciences and Engineering Council of Canada (NSERC) through its Discovery Grant program. We gratefully acknowledge the support of Fonds de recherche du Québec through their strategic infrastructure grants.

References

K. An and H. K. Lo. Service reliability–based transit network design with stochastic demand. *Transportation Research Record*, 2467(1):101–109, 2014.

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

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

J. Andersen, M. Christiansen, T.G. Crainic, and R. Grønhaug. Branch and price for service network design with asset management constraints. *Transportation Science*, 46 (1):33–49, February 2011.

C. Archetti, M. G. Speranza, and M. W. P. Savelsbergh. An optimization-based heuristic for the split delivery vehicle routing problem. *Transportation Science*, 42(1):22–31, 2008.

C. Archetti, A. Corberan, I. Plana, J. M. Sanchis, and M. S. Grazia. A matheuristic for the team orienteering arc routing problem. *European Journal of Operational Research*, 245(2):392–401, 2015.

D. Bertsimas and J. N. Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.

W. Cassidy. LTL carriers pave way to growth in changing market. *Journal of Commerce*, March, 2017.

M. Chouman and T.G. Crainic. Cutting-Plane Matheuristic for Service Network Design with Design-Balanced Requirements. *Transportation Science*, 49(1):99–113, 2015.

I. Contreras and E. Fernandez. General network design: A unified view of combined location and network design problems. *European Journal of Operational Research*, 219(3):680 – 697, 2012. URL `http://www.sciencedirect.com/science/article/pii/S0377221711009969`.

T. H. Cormen. *Introduction to algorithms.* MIT press, 2009.

T. G. Crainic, M. Hewitt, M. Toulouse, and D. M. Vu. Scheduled service network design with resource acquisition and management. *EURO Journal on Transportation and Logistics*, 7(3):277–309, 2017.

T.G. Crainic. Network Design in Freight Transportation. *European Journal of Operational Research*, 122(2):272–288, 2000.

T.G.. Crainic, X. Fu, M. Gendreau, W. Rei, and S. Wallace. Progressive hedging-based metaheuristics for stochastic network design. *Networks*, 58(2):114–124, 2011.

T.G. Crainic, M. Hewitt, and W. Rei. Scenario grouping in a progressive hedging-based meta-heuristic for stochastic network design. *Computers & Operations Research*, 43 (2):90–99, 2014a.

T.G. Crainic, M. Hewitt, M. Toulouse, and D.M. Vu. Service Network Design with Resource Constraints. *Transportation Science*, 50(4):1380–1393, 2014b.

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

FMCSA. Summary of hours of service regulations, December 2014. URL http://www.fmcsa.dot.gov/regulations/hours-service/summary-hours-service-regulation

M. Hewitt, G.L. Nemhauser, and M.W.F. Savelsberg. Combining exact and heuristic approaches for the capacitated fixed-charge network flow problem. *INFORMS Journal on Computing*, 22(2):314–325, 2010.

K. Høyland, M. Kaut, and S. W. Wallace. A heuristic for moment-matching scenario generation. *Computational Optimization and Applications*, 24:169–185, 2003.

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

D. Kim, C. Barnhart, K. Ware, and G. Reinhardt. Multimodal express package delivery: a service network design application. *Transportation Science*, 33(4):391–407, 1999.

M. F. Lai and H. Lo. Ferry service network design: optimal fleet size, routing, and scheduling. *Transportation Research Part A: Policy and Practice*, 38(4):305–328, 2004.

A.-G. Lium, T.G. Crainic, and S.W. Wallace. A study of demand stochasticity in service network design. *Transportation Science*, 43(2):144–157, 2009.

S. Melkote and M. Daskin. Capacitated facility location/network design problems. *European Journal of Operational Research*, 129(3):481 – 495, 2001a. ISSN 0377-2217. doi: 10.1016/S0377-2217(99)00464-6. URL http://www.sciencedirect.com/science/article/pii/S0377221799004646.

S. Melkote and M. Daskin. An integrated model of facility location and transportation network design. *Transportation Research Part A: Policy and Practice*, 35(6): 515 – 538, 2001b. ISSN 0965-8564. doi: 10.1016/S0965-8564(00)00005-7. URL http://www.sciencedirect.com/science/article/pii/S0965856400000057.

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

W. B. Powell. A local improvement heuristic for the design of less-than-truckload motor carrier networks. *Transportation Science*, 20(4):246–257, 1986.

V. Schmid, K. F. Doerner, R. F. Hartl, M. W. P. Savelsbergh, and W. Stoecher. A hybrid solution approach for ready-mixed concrete delivery. *Transportation Science*, 43(1):70–85, 2009.

K. Smilowitz, A. Atamtürk, and C. Daganzo. Deferred item and vehicle routing within integrated networks. *Transportation Research Part E: Logistics and Transportation Review*, 39(4):305–323, 2003.

UPS. UPS fact sheet. Technical report, 2019. https://www.pressroom.ups.com/.

J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco. A matheuristic for the truck and trailer routing problem. *European Journal of Operational Research*, 230 (2):231–244, 2013.

D.M. Vu, T.G. Crainic, and M. Toulouse. A Three-Stage Matheuristic for the Capacitated Multi-commodity Fixed-Cost Network Design with Design-Balance Constraints. *Journal of Heuristics*, 19:757–795, 2013.

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