CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Integrating Resource Acquisition and Management Decisions Into Tactical Transportation Planning Under Uncertainty

Mike Hewitt
Teodor Gabriel Crainic
Maciek Nowak
Walter Rei

August 2017

CIRRELT-2017-52

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Integrating Resource Acquisition and Management Decisions Into Tactical Transportation Planning Under Uncertainty

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. This paper focuses 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 propose 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, the model takes the form of a stochastic program explicitly addressing uncertainty in demand through the use of scenarios. 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 problems. Extensive computational experiments show that it is effective, and validate the solutions by analyzing their attributes as instance parameters vary.

Keywords: Consolidation freight carrier planning, scheduled service network design with resource management, resource acquisition and allocation, 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.

---

* Corresponding author: mhewitt3@luc.edu

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2017

1 Introduction

This research is focused on modeling and solving a planning problem faced by freight transportation companies; specifically, consolidation carriers. Consolidation carriers transport customer shipments that are small relative to the vehicle or convoy capacity and have enabled, amongst other things, the liberalization and exponential growth of world trade of consumer goods and the transformative effects of eCommerce. The 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, are all based on consolidation. 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 $51 billion in revenue in 2016 (UPS, 2017). Consolidation carriers play a prominent role in the fulfillment of orders placed online, in brick-and-mortar stores, and through other channels.

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 the first model and solution method for acquiring and locating resources while designing a service network and managing the resources necessary to execute it. However, that model presumes that customer demands are known with certainty.

Thus, to reflect industry concerns and the fact that the true cost of a fleet includes both the money required to acquire it and the cost of the transportation plans it enables, we propose 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. As a well-sized fleet should be able to accommodate fluctuations in freight volumes, the proposed model explicitly represents uncertainty in freight volumes. Specifically, demand uncertainty is modeled through the use of scenarios and the model is in fact a stochastic program. Solving this program will assist transportation companies 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 propose a matheuristic for addressing 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. This paper thus presents matheuristic-based solution methods for a stochastic programming network-design model applicable to the considered strategic and tactical planning problem that is faced by numerous transportation companies.

The proposed matheuristic takes the form of a neighborhood-based search scheme, where parts of the search space (variables) is fixed and the resulting restricted mix-integer problem is solved exactly. This space-decomposition idea follows the successful contributions of Hewitt et al. (2015), for vehicle routing, and Hewitt et al. (2010) and Erera et al. (2013) for LTL service network design. To our best knowledge, this idea has never been developed for stochastic (service) network design.

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). 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 suggest that it is effective.

The paper is organized as follows. We next, Section 2, 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, Section 4, while Section 5 is dedicated to the solution method. e describe the experimental design in Section 6, and present the results and analyzes in Section 7. We conclude in Section 8.

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 the strategic decisions can have a profound impact on the cost of transporting freight, we propose to solve them jointly. In this way, and by explicitly modeling uncertainty in freight volumes, 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*, the resulting transportation plan, selected services and resources, to be repeatedly executed during the *tactical planning horizon* length. During this period of time, the carrier must transport a set of customer shipments. The geographic (origin and destination locations) and temporal (availability time and due arrival time) attributes of customer shipments are assumed as known. However, to ensure decisions (both strategic and tactical) that are robust with respect to customer demands, only the probability distributions for freight volumes are presumed as known. In this sense, this research extends the work presented in Crainic et al. (2017), wherein only a point forecast of customer demands was used, and thus a deterministic model solved.

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. 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. 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 so that it may be possible to perform a better 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.

We only presume that the execution of a service requires a resource, not 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 service/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 extend the work presented in Crainic et al. (2014b, 2017) by integrating strategic decisions into the model. Specifically, the number of resources to use, and the home terminal of each. However, 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 schedule (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.

As this paper explicitly models uncertainty in customer demands (i.e., freight volumes), 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 are considering a situation wherein a carrier signs a long-term contract with a third party carrier. The carrier may take two recourse actions after volumes are known: (1) to outsource the delivery of an individual shipment from origin to destination to 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 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.

Attention next turns to 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).

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 brief 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. Moreover, to our best knowledge, none of these integrates demand uncertainty and how it affects the design of the service and the deployment of resources. Our contribution to addressing this issue follows.

4 The Model - A Path-based Formulation

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. However, 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. We note that this work extends that of Crainic et al. (2017), and thus 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 $(t + \pi_{lm}) \bmod TMAX$. 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 $(l_t, l_{(t+1) \bmod TMAX})$ 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 t and must be transported to terminal m by period t' as a commodity with index k, origin node $o(k) = l_t$, and destination node $d(k) = m_{t'}$. 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 as known, we assume there is uncertainty in their volume. A set of scenarios Ψ model uncertainty in demand volumes. As such, the value $q^{k\psi\psi}$ represents the volume of commodity $k \in \mathcal{K}$ in scenario $\psi \in \Psi$. Let $q_i^{k\psi} = q^{k\psi}$ when $i = o(k)$, $q_i^{k\psi} = -q^{k\psi}$ when $i = d(k)$, and $q_i^{k\psi} = 0$ otherwise. Finally, we associate with each scenario ψ the probability ϕ_ψ.

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 commoditiy'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, presume one path 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.

Having presented the model for tactical, service network design-type decisions, attention next turns to how to model the strategic, resource-related decisions the planner must make. Conceptually, a "source" layer in the time-space network models the 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 acquision 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.

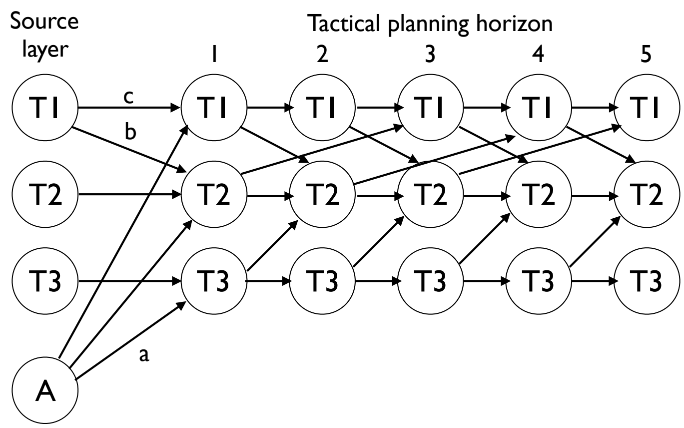


Figure 1: Modeling strategic and tactical decisions (not all arcs included)

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.

Ultimately, we seek to solve what we call the *Scheduled Service Network Design with Resource Acquisition and Management under Uncertainty* (*SSND-RAMU*) problem, which in its scenario-based deterministic forms aims to

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

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

subject to

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

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

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

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

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

$$y_{l_tm_{t'}} \le \sum_{h\in\Lambda}\sum_{\tau\in\theta_h} r^\tau_{l_tm_{t'}} z_h^\tau, \quad \forall (l_t,m_{t'})\in\mathcal{S}, \tag{8}$$

$$x_p^{k\psi} \ge 0, \quad \forall p\in\mathcal{P}_k^\psi,\ k\in\mathcal{K},\psi\in\Psi, \tag{9}$$

$$a_{wh}\in Z, \quad \forall w\in\Lambda^+, h\in\Lambda, \tag{10}$$

$$z_h^\tau\in\{0,1\}, \quad \forall h\in\Lambda,\tau\in\theta_h, \tag{11}$$

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

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

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 (9), (10), (11), (12), and (13) define the decision variables of the model and their domains.

5 Solution Approaches

The model consists of two sets of variables, paths and cycles, that are typically too large to be enumerated *a priori*. The two solution approaches we present for the SSND-RAMU thus integrate the dynamic generation of these variables. The first solution approach is a standard column generation-based (CG; Bertsimas and Tsitsiklis, 1997) heuristic (Section 5.2). This method serves as a benchmark for the second approach presented, a matheuristic (Section 5.3). As both heuristics presented 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$_{LPR}$

The proposed solution methods dynamically generate path, $x_p^{k\psi}$, and resource cycle variables z_h^{τ}. Let's assume we have 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 $\bar{\theta} \subseteq \theta = \cup_{h \in \Lambda} \theta_h$. Then, define SSND-RAMU$(\bar{\theta}, \bar{\mathcal{P}})$ (and its linear programming relaxation SSND-RAMU$(\bar{\theta}, \bar{\mathcal{P}})_{LPR}$) as 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$(\bar{\theta}, \bar{\mathcal{P}})_{LPR}$, 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). We next describe how we use these duals to generate paths and cycles.

Generating paths: With these dual variables, we have the following formula for the reduced cost ($\bar{c}_p^{k\psi}$) associated with commodity k using path $p \in \mathcal{P}_k^{\psi}$ in scenario $\psi \in \Psi$, $\bar{c}_p^{k\psi} = c_p^k - \alpha_k^{\psi} - \sum_{(l_t, m_{t'}) \in p} q^{k\psi} \beta_{l_t m_{t'}}^{\psi}$. Thus, after having solved SSND-RAMU$(\bar{\theta}, \bar{\mathcal{P}})_{LPR}$, we seek to find paths such that $\bar{c}_p^{k\psi} < 0$, or, such that $\sum_{(l_t, m_{t'}) \in p} (c_{l_t m_{t'}}^k - q^{k\psi} \beta_{l_t m_{t'}}^{\psi}) < \alpha_k^{\psi}$. 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 $c^k_{l_t m_{t'}} - q^{k\psi}\beta^{\psi}_{l_t m_{t'}}$. 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 ($\bar{O}^{\tau}_h$) associated with having a resource that is assigned to terminal h follow the itinerary dictated by cycle $\tau \in \theta_h, \bar{O}^{\tau}_h = O^{\tau}_h - \xi_h + \sum_{(l_t,m_{t'})\in\tau} \gamma_{l_t m_{t'}}$. Thus, variables should be determined such that $\bar{O}^{\tau}_h < 0$, or, such that $\sum_{(l_t,m_{t'})\in\tau} \gamma_{l_t m_{t'}} < \xi_h - O^{\tau}_h$. Formally, given a home terminal h, we seek to solve the optimization problem

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

subject to

$$\tau \in \theta_h. \tag{14}$$

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. We thus seek to solve 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

$$\sum_{t=2}^{TMAX} \eta_t \geq \kappa \tag{15}$$

$$\eta_{TMAX} = 1, \tag{16}$$

$$\sum_{(h_1,m_{t'})\in\mathcal{A}} v_{h_1,m_{t'}} = 1, \tag{17}$$

$$\sum_{(l_{t'},h_t)\in\mathcal{A}} v_{l_{t'},h_t} = \eta_t, \tag{18}$$

$$\sum_{(m_{t'},l_t)\in\mathcal{A}} v_{m_{t'},l_t} - \sum_{(l_t,n_{t''})} v_{l_t,n_{t''}} = 0 \quad \forall l_t \in \mathcal{N}, \tag{19}$$

$$v_{l_t m_{t'}} \in \{0,1\} \quad \forall (l_t, m_{t'}) \in \mathcal{A} \tag{20}$$

$$\eta_t \in \{0,1\} \quad \forall t = 1,\ldots,TMAX \tag{21}$$

Constraints (15) ensure that the cycle returns to the home terminal, h, at least k times, whereas constraints (16) ensure that the home terminal is returned to at the end of the planning horizon. Similarly, constraints (17) ensure that the cycle begins at the home terminal, h. Then, constraints (18) link the movements in the cycle to the periods when it must return home, and constraints (19) ensure that the movements can be decomposed into cycles. Finally, constraints (20) and (21) define the variables of the model and their domains. Having solved Price-Cycle(h, γ), if $R_h < \xi_h - O_h^\tau$, then a cycle with negative reduced cost exists which joins the set $\bar{\theta}_h$.

Algorithm 1 solves the SSND-RAMU$(\bar{\theta}, \bar{\mathcal{P}})_{LPR}$. Let $\theta_{LPR}, \mathcal{P}_{LPR}$ be the sets of cycles and paths generated by the algorithm at termination. Similarly, let $\theta^*_{LPR}, \mathcal{P}^*_{LPR}$ be the sets of paths and cycles used in the final solution to SSND-RAMU$(\bar{\theta}_l, \bar{\mathcal{P}}_k)_{LPR}$ produced by Algorithm 1. 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 observe that at an iteration of a column generation algorithm a dual bound can be produced on the optimal value of the linear programming problem, thus producing an optimality gap. As a result the algorithm is also terminated when that optimality gap is within a pre-specified tolerance, ϵ.

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

Initialize $\bar{\mathcal{P}}_k^\psi$ 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 $\bar{\mathcal{P}} = \bigcup_{k \in \mathcal{K}, \psi \in \Psi} \bar{\mathcal{P}}_k^\psi$
while stopping criteria not met **do**
  Solve SSND-RAMU$(\bar{\theta}, \bar{\mathcal{P}})_{LPR}$ for dual variables $\xi_h, \alpha_k^\psi, \beta^\psi_{l_t m_{t'}}, \gamma_{l_t m_{t'}}$
  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 $< \alpha_k^\psi$ then add path to $\bar{\mathcal{P}}_k^\psi$.
  end for
  $\bar{\mathcal{P}} = \bar{\mathcal{P}} \bigcup_{k \in \mathcal{K}, \psi \in \Psi} \bar{\mathcal{P}}_k^\psi$
  for all $h \in \Lambda$ **do**
    Solve Price-Cycle(h, γ) for value R_h and cycle τ
    if $R_h < \xi_h - O_h^\tau$ **then**
      Add τ 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 (θ_{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*. Algorithm 3 presents a formal description of the methodology, its steps being then described in greater detail.

Algorithm 3 IP-based Mathheuristic (IP-Solve)

Solve SSND-RAMU$_{LPR}$ via column generation for cycles θ^*_{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}.

The solution of a restricted instance of SSND-RAMU$(\theta_{cand}, \mathcal{P}_{cand})_{LPR}$ (with column generation) generates new cycles and paths. To restrict the instance, we first partition 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, fixing the value of the variable z^{τ}_{h} to 1(0) when $\tau \in \theta^{one}_{cand}$ ($\tau \in \theta^{zero}_{cand}$) creates a partial solution to SSND-RAMU$(\theta_{cand}, \mathcal{P}_{cand})_{LPR}$. Algorithm 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,

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

that measures the expected utilization of services in that cycle in the current solution. Cycles in $\theta_{cand} \setminus \theta_{sol}$ are assigned a score, σ_τ, of zero. 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.

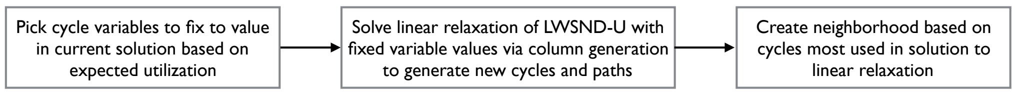


Figure 2: *CG-Nbhd*: A neighborhood based on generating cycles and paths

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. Specifically, each scenario ψ has a set of paths, $\mathcal{P}^{\psi}_{nbhd}$, created that can be taken by a commodity. Thus, the selection of a subset of scenarios, $\bar{\Psi} \subseteq \Psi$, wherein $|\bar{\Psi}|$ is an algorithm parameter, creates this neighborhood. Then, for $\psi \in \Psi \setminus \bar{\Psi}$, $\mathcal{P}^{\psi}_{nbhd}$ includes paths p wherein $\bar{x}^{kw}_p > 0$, whereas $\mathcal{P}^{\psi}_{nbhd} = \mathcal{P}^{\psi}_{cand}$ for $\psi \in \bar{\Psi}$. In other words, for $\psi \in \Psi \setminus \bar{\Psi}$, commodities are restricted to the paths they follow in the current solution, whereas for $\psi \in \bar{\Psi}$ a commodity can follow any known path. The set $\bar{\Psi}$ is determined randomly.

6 Experimental Design

We next describe the experiments used to validate the model, SSND-RAMU, and the solutions produced by IP-Solve. We first discuss the transportation network that all instances are based on. We then discuss the distribution for freight volumes derived from data from a Less-than-truckload transportation (LTL) carrier in the United States and describe how scenarios model that distribution. We finish with a discussion of the values used for model parameters and how they were derived.

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 ($|\mathcal{S}| = 600$).

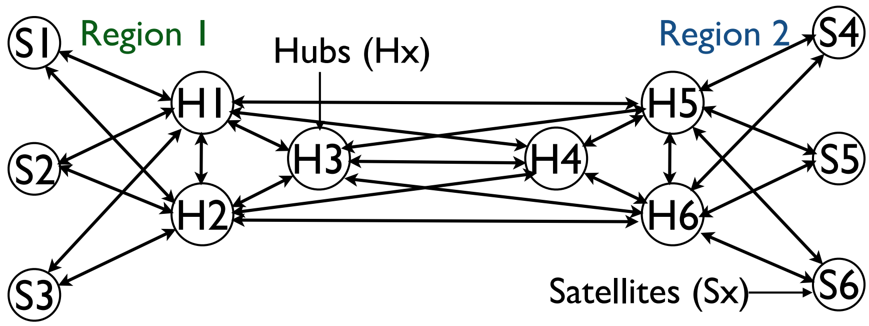


Figure 3: Hub and spoke network used in experiments

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 1 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 1: Shipments in instance

Attention next turns to the statistical distribution used for freight volumes. From one week's worth of data from a US carrier, we derived that the volumes could be approximated with a General Beta distribution, of the form $d = 150 + 21,050Z$ wherein

Z followed a beta distribution with parameters $a = .057706$ and $b = 7.79$. It was then presumed that the volume of all 228 commodities followed this same distribution, and that their volumes were independent. Then, given these distributions, we determined the four moments of the distribution and used the algorithm described in Høyland et al. (2003) to generate 24 scenarios to represent their joint distribution. 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. The smallest number of scenarios that yielded a close approximation of the first four moments was 24.

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 $F_{wh}, w \in \Lambda$.

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 2. 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 2: Cost structures for acquiring resources

Finally, we 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 ($m_\sigma = 3$). Table 3 summarizes the instance parameter values and their variation.

In summary, the problem set consists of 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.

Parameter	Values considered	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 3: Instance parameter values

7 Computational Study

An extensive computational study tested the effectiveness of IP-Solve. In all experiments, we executed both algorithms (CG-Solve and IPS) 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%. A time limit of 10 minutes bounded the solution of SSND-RAMU$(\bar{\theta}, \bar{\mathcal{P}})_{LPR}$ at the beginning of CG-Solve and IP-Solve. 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 our experiments and present those values in Table 4.

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

Table 4: Algorithm parameter values

We divide our discussion into two sections: (1) studying the ability of IP-Solve to produce high-quality solutions and understanding its behavior, and, (2) validating the use of the SSND-RAMU model and the solutions produced by IP-Solve.

7.1 Analyzing the performance of IP-Solve

As we execute both algorithms on each instance, each instance has two objective function values: (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. Finally, we also consider the objective function value, $obj_{CG-Solve}^{IP-Solve-TTB}$, of the best solution found by CG-Solve by the time IP-Solve found its best solution. The analysis of the performance of IP-Solve begins by calculating two gaps, $gap_{CG-Solve}^{IP-Solve} = (obj_{CG-Solve} - obj_{IP-Solve})/obj_{CG-Solve}$, and

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

First, note that for every instance $gap^{IP-Solve}_{CG-Solve}$ is positive, meaning IP-Solve produced a better solution than CG-Solve. Also note that the average of $gap^{IP-Solve}_{CG-Solve}$ over all instances is 5.44% and the average of $gap^{IP-Solve-TTB}_{CG-Solve}$ is 5.62%. To get a clearer understanding of the relative performance of the two algorithms, Figure 4a presents the distributions of these gaps (e.g., the percentage of instances for each % wherein the gap is in that range). 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.

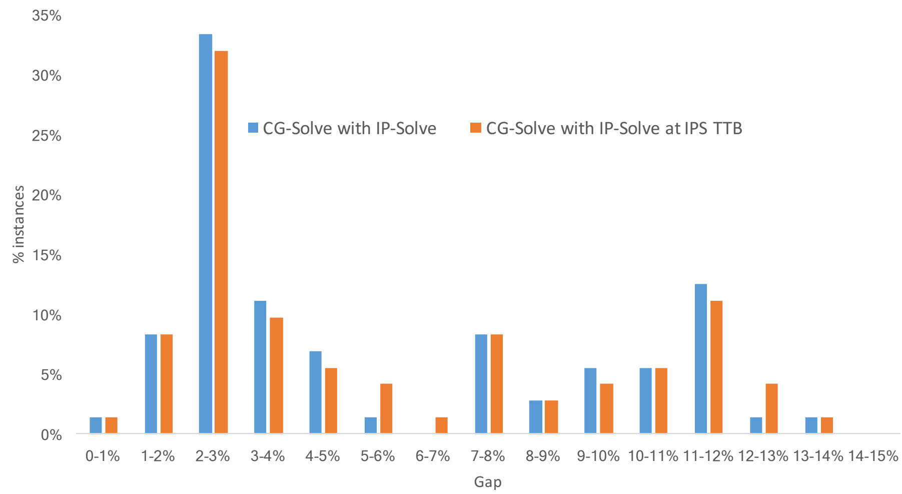
Focusing on the length of time required by IP-Solve to find its best solution and the improvement of that solution over the first one found provides additional insight. Recall that the first solution is found by solving a MIP wherein only the paths and cycles that appear in the solution to SSND-RAMU$(\bar{\theta}, \bar{\mathcal{P}})_{LPR}$ produced by Algorithm 1 are considered. Averaging over all instances, it takes IP-Solve 3,339.43 seconds (roughly 56 minutes) to find its best solution. The gap between these solutions is $gap^{best}_{first} = (obj_{first} - obj_{best})/obj_{first}$, 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, the best solution is 7.17% better than the initial one found (i.e., $gap^{best}_{first} =$ 7.17%). Figures 4b and 4c complement these summary statistics with the distributions for each statistic. Figure 4b shows that IP-Solve often uses nearly all the time allotted to find its best solution, suggesting it is thoroughly searching the solution space. Figure 4c indicates that IP-Solve improves upon the initial solution by more than 5% in over half the instances.

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 5 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.

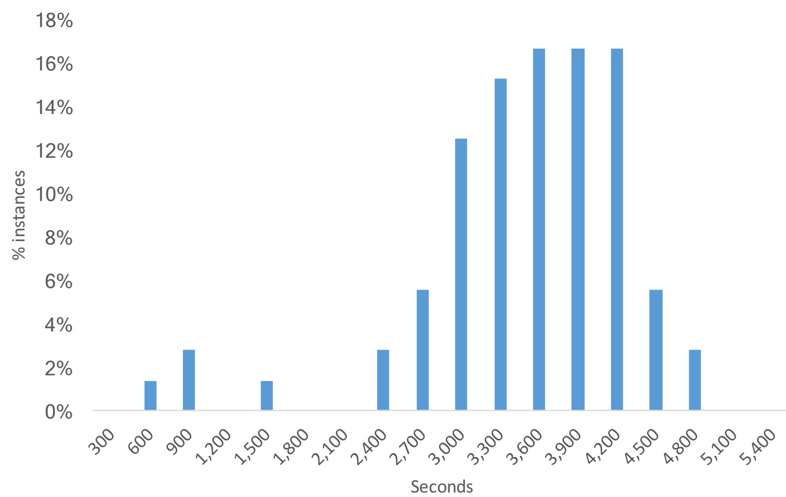
Table 5: Neighborhood statistics

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

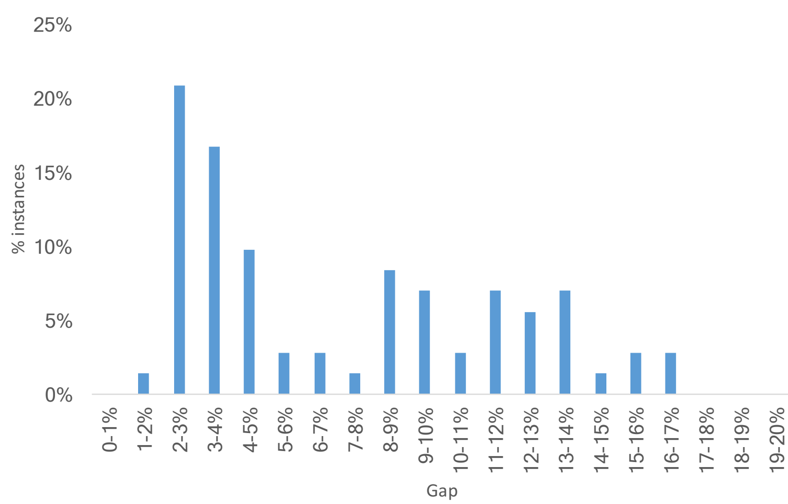
Attention next turns to the sensitivity of the algorithm to the instance parameters reported in Table 3. To measure this, we average $gap^{IP-Solve}_{CG-Solve}$ over all instances with the same parameter value (e.g., all instances wherein $\mu^e = 3$). Table 6 presents results according to cost-based parameters, while Table 7 presents results according to demand-based



(a) IP-Solve vs CG-Solve



(b) IP-Solve times to best solution



(c) gap_{first}^{best}

Figure 4: Comparing Algorithms - Gap Distributions

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 6: 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
$gap^{IP-Solve}_{CG-Solve}$	5.22%	5.87%	5.72%	5.29%	6.20%	6.58%	2.58%	6.38%

Table 7: Algorithm performance by freight demand-based instance parameters

Parameter	Volume v_d		Variation m_σ		
Value	1	5	1	2	3
$gap^{IP-Solve}_{CG-Solve}$	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 $obj_{CPLEX-5hours}$ 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 IPS and the best solution produced by CPLEX after solving SSND-RAMU(θ_{cand}, $\mathcal{P}_{cand}$), calculated as $primal^{CPLEX-5h}_{IP-Solve} = (obj_{IP-Solve} - obj_{CPLEX-5hours})/obj_{IP-Solve}$. We also calculate the optimality gap associated with the solution produced by IP-Solve and the dual bound found by CPLEX, $opt^{CPLEX-5h}_{IP-Solve} = (obj_{IP-Solve} - bound_{CPLEX-5hours})/obj_{IP-Solve}$. Table 8 reports averages of these gaps over all instances with the same volume and variation multiplier.

Table 8: Comparison with CPLEX solving SSND-RAMU($\theta_{cand}, \mathcal{P}_{cand}$) for five hours

	$primal^{CPLEX-5h}_{IP-Solve}$				$opt^{CPLEX-5h}_{IP-Solve}$			
	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%

Observation of the $primal^{CPLEX-5h}_{IP-Solve}$ results indicates 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 $opt^{CPLEX-5h}_{IP-Solve}$ 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. Attention can next turn to validating the use of SSND-RAMU and the solutions produced by IP-Solve.

7.2 Validation

To validate the model, SSND-RAMU, we first calculate the *Value of the Stochastic Solution* (VSS). Then, to validate the solutions produce by IP-, we look at how instance parameters impact the allocation of resources to terminals and outsourcing decisions.

The mean demand for each product, $\bar{q} = \sum_{\psi \in \Psi} \phi_w q^{k\psi}$, allows for the calculation of the mean-value scenario, $\bar{\psi}$, which in turn leads to the VSS. CG-Solve executes SSND-RAMU, formulated with just the mean-value scenario $\bar{\Psi}$ and with the MIP SSND-RAMU($\theta_{LPR}, \mathcal{P}_{LPR}$) solved for ten hours. Solving this MIP 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 ($\bar{y}^{e}_{l_t,m_{t'}}$). We then evaluate these decisions 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 execute CG-Solve, 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 the gap between these two costs, $(obj_{mean} - obj_{SSND-RAMU})/obj_{mean}$. Table 9 reports averages of these gaps, by variation (m_σ) and cost structure.

Table 9: 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%

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

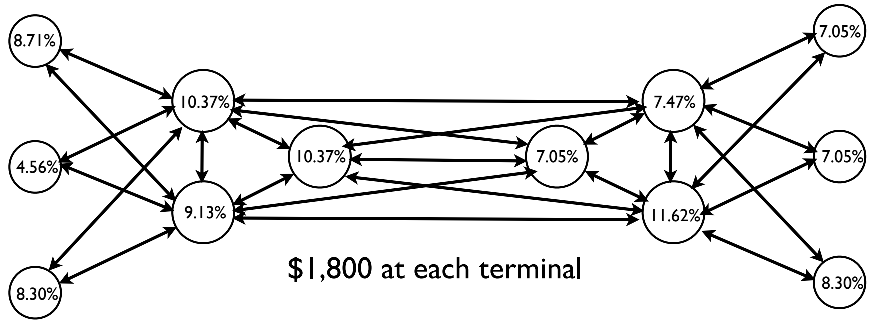
We next study the allocation of resources to terminals under each of the four cost structures described in Table 2 (and repeated in each Figure). Figures 5a, 5b, 5c, and 5d report the percentage of resources allocated to each terminal, calculated as $100 * \frac{p_{Ah}}{\sum_{h' \in \Lambda} p_{Ah'}}, h \in \Lambda$.

Examining these figures, we conclude that both freight volumes and resource acquisition costs that vary by terminal can have an impact on where resources should be allocated. Focusing on Figure 5a, wherein resource acquisition costs are the same for all terminals, the distribution of resources is not even (with a greater proportion of resources assigned to hubs than satellites). In this case freight volumes drive resource allocation decisions. However, the remaining cost structures, wherein acquisition costs vary by terminal, see a drastic change in the distribution of resources. For example, in Figure 5c, wherein it is much cheaper to acquire resources for a hub, no resources are acquired for satellites. And, in the (somewhat pathological) fourth cost structure (Figure 5d), the distribution of resources is correlated with the acquisition costs. For example, over 85% of the resources acquired are assigned to one of the four cheapest terminals (S1,H1,H3, and S6), in terms of acquisition costs. These results indicate that modeling both (strategic) resource acquisition and allocation and (tactical) service network design decisions can lead to better plans. They also indicate that IP-Solve is producing solutions that exhibit expected traits. Thus, we conclude that IP-Solve is producing high-quality solutions from this perspective as well.

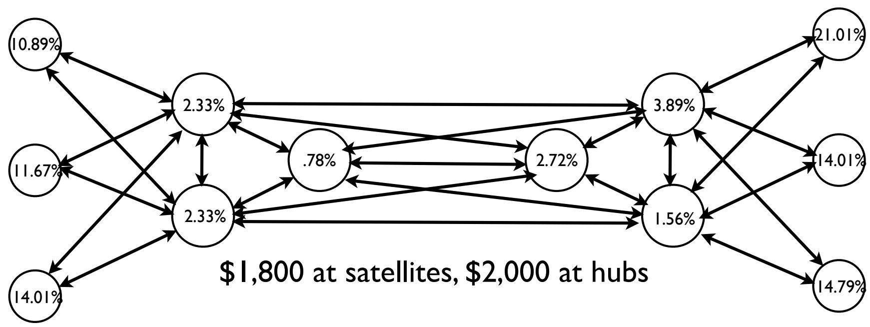
We next turn our attention to how the outsource multipliers, μ^e, μ^σ impact outsourcing and resource acquisition decisions. To that effect, given the best solution produced by IP-Solve, and its objective function value obj, we calculate the following statistics: (1) $\sum_{(l_t,m_{t'})\in\mathcal{S}} f^e_{lm} y^e_{l_t m_{t'}}/obj$ (Out), (2) $\sum_{s\in\Psi} p_s \sum_{(l_t,m_{t'})\in\mathcal{S}} f^\sigma_{lm} y^{\sigma s}_{l_t m_{t'}}/obj$ (Spot), and, (3) $h_{acq} = \sum_{h\in\Lambda} p_{Ah}$ (Res-acq). The first statistic represents the percentage of costs that can be attributed to outsourcing the execution of services on a long-term contract basis. The second statistic is similar, albeit looks at the expected amount spent outsourcing on the spot market. The last measures the total number of resources acquired.

Table 10 reports these statistics, for different values of μ^e, μ^σ and both baseline and higher freight volumes. However, the table reports the values of obj and Res-acq relative to the values seen in the solution with $\mu^e = 2, \mu^\sigma = 3$ for each demand volume, as absolute numbers provide no information in this context. For example, for $v_d = 1$, an increase in the spot market outsourcing cost (from $\mu^\sigma = 3$ to 4) increased the objective function value by 2.37% and lead to a 8.21% increase in the number of resources acquired.

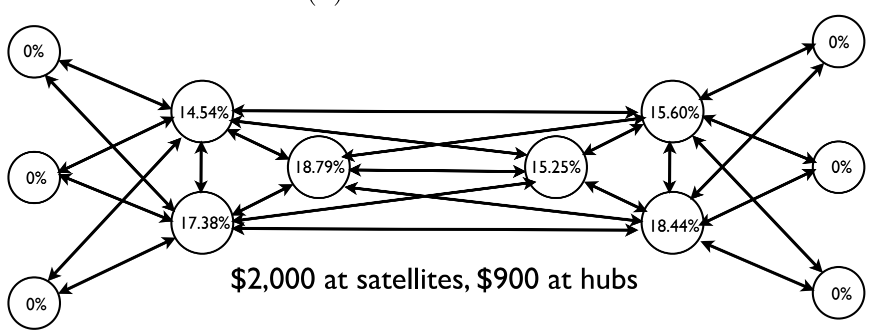
Both demand levels follow the same pattern. First, note that the total cost of the



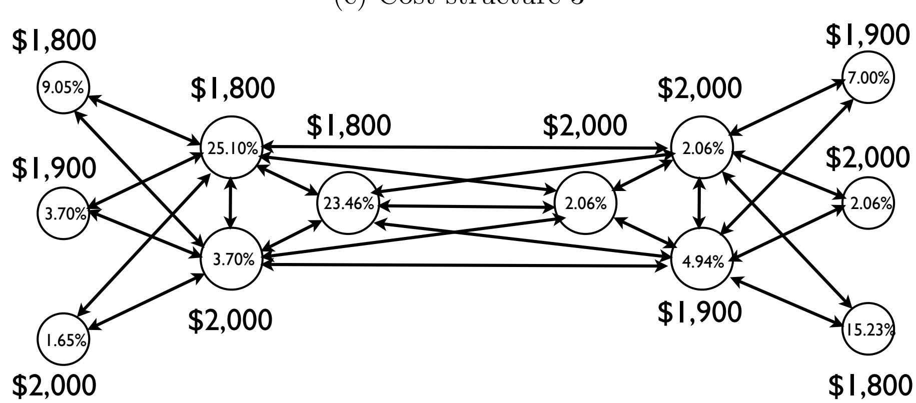
(a) Cost structure 1



(b) Cost structure 2



(c) Cost structure 3



(d) Cost structure 4

Figure 5: Resource allocation for different cost structures

	μ^e	$\mu^\sigma = 3$				$\mu^\sigma = 4$			
		Out	Spot	*obj*	Res-acq	Out	Spot	*obj*	Res-acq
$v_d = 1$	2	3.80%	3.89%	100%	100%	3.84%	2.52%	97.63%	108.94%
	3	-	-	-	-	0.65%	2.70%	105.30%	128.86%
$v_d = 5$	2	1.54%	1.20%	100%	100%	2.11%	0.94%	100.09%	96.44%
	3	-	-	-	-	0.65%	2.70%	101.53%	131.67%

Table 10: Outsourcing and acquisition decisions

solutions produced by IP-Solve does not vary significantly as the cost parameters change (at most a 5.03% increase when both long-term and spot market outsourcing costs increase). We attribute this to the increase in resources acquired and hypothesize that IP-Solve is adapting to the cost structure with which it is presented. In a sense, this is one advantage of using a matheuristic that explicitly captures the impact on the objective of moving to a neighboring solution. Also, as outsourcing on the spot market becomes more expensive (e.g., $\mu^\sigma = 4$), more is spent on long-term, contract outsourcing. Yet, the number of resources remains relatively constant. However, as both types of outsourcing become more expensive, the algorithm adapts by acquiring more resources. This last point is another validation for modeling resource acquisition and service network design decisions in an integrated manner.

8 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 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 show that it is effective, the second matheuristic, called IP-Solve, being superior. We also validated the solutions by analyzing their attributes as instance parameters vary.

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.

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

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. 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.

T.G. Crainic, M. Hewitt, M. Toulouse, and D.M. Vu. Service Network Design with Resource Acquisition and Management. *EURO Journal on Transportation and Logistics*, 2017. forthcoming.

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-regulations.

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.

M. Hewitt, M. Nowak, and L. Gala. Consolidating home meal delivery with limited operational disruption. *European Journal of Operational Research*, 243(1):281–291, 2015.

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, 2017. 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.