CIRRELT

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

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

# Scheduled Service Network Design with Resource Acquisition and Management

**Teodor Gabriel Crainic**
**Mike Hewitt**
**Michel Toulouse**
**Duc Minh Vu**

**October 2016**

**CIRRELT-2016-50**

**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**

# Scheduled Service Network Design with Resource Acquisition and Management

**Teodor Gabriel Crainic[1,2,*], Mike Hewitt[3], Michel Toulouse[1,4], Duc Minh Vu[3]**

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Management and Technology, Université du Québec à Montréal, P.O. Box 8888, Station Centre-Ville, Montréal, Canada H3C 3P8

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

[4] Vietnamese-German University, Le Lai Street, Hoa Phu Ward, Binh Duong New City, Binh Duong Province, Vietnam

**Abstract.** We present a new planning model for freight consolidation carriers, one that links strategic resource-acquisition and allocation decisions with tactical, service network design related decisions. Specifically, like service network design models that recognize resource constraints, the model selects services and routes both commodities and the resources needed to support the services that transport them. In addition, the model recognizes that resources can be grouped into types that differ from one another with respect to capabilities, e.g. speeds, capacities, scheduling rules, etc. Ultimately, along with recognizing resource constraints, the model also makes strategic decisions such as how many resources of each type should be acquired, to what terminal new resources should be assigned, and which existing terminal-based resources should be reassigned. As such, the model can be used from a strategic planning, resource-acquisition, mixing, and allocation perspective as it provides an estimate of the impact of such decisions on transportation costs. We extend a solution approach for a service network design problem with a fixed set (both in number and allocation) of resources of a single type to one that can also make these acquisition and allocation decisions for multiple types of resource. Then, with an extensive computational study, we demonstrate the efficacy of the approach and benchmark its performance against both a leading commercial solver and a column generation-based heuristic. Finally, we perform an extensive computational study to understand how the resource-related and service network design-related components of the model interact, including how freight volumes and cost structures impact how many resources should be acquired.

**Keywords**: Service network design, fleet sizing and management, matheuristics, slope scaling, column generation.

**Acknowledgements.** While working on this project, the fourth author was doctoral student with Computer Science and Operations Research Department, Université de Montréal, and student member of CIRRELT. Partial funding for this project has been provided by the Natural Sciences and Engineering Research Council of Canada (NSERC), through its Discovery Grant program and by the EMME/2-STAN Royalty Research Funds (Dr. Heinz Spiess). We also gratefully acknowledge the support of Fonds de recherche du Québec – Nature et technologie (FRQNT) through their infrastructure grants and of Calcul Québec and Compute Canada through access to their high-performance computing infrastructure.

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: TeodorGabriel.Crainic@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2016

© Crainic, Hewitt, Toulouse, Minh Vu and CIRRELT, 2016

# 1 Introduction

Consolidation carrier is an umbrella term for companies that specialize in transporting customer shipments that are small, relative to vehicle capacity, or have low value, relative to transportation costs, or both, to be economical to transport them directly from customer origin to customer destination. These types of carriers participate in nearly every segment of the transportation industry, including small-package transportation companies, Less-than-Truckload (LTL) motor carriers, railroads, ocean/maritime liner navigation, and land and water-based intermodal transportation. And each of these industries is large. For example, LTL freight is a roughly $30 billion industry in the United States, and small package is even larger, with one player alone (UPS) reporting $54 billion in revenue in 2012. To achieve these revenues, carriers in these industries run high volume operations; LTL carriers often spend millions of dollars in transportation and handling costs each week, and UPS reportedly delivered 16.3 million packages and documents a day in 2012.

In addition, the efficiency and timeliness with which consolidation carriers operate has a direct impact on the profitability of many other types of firms. They support domestic and global supply chains, enabling product manufacturers to provide their products in a timely and low-cost manner to their customers, be they manufacturers, wholesalers, or distributors. Consolidation carriers also play a prominent role in the fulfillment of product orders placed online, in brick-and-mortar stores, and through other channels.

Consolidation carriers reduce their transportation costs through economies of scale. To achieve this, carriers route shipments through a network of terminals wherein they consolidate shipments from various customers and with various origins and destinations into the same vehicles. Transportation between two terminals in the network, with or without intermediate stops, is often referred to as a *service*. The primary cost associated with a service is the fixed cost incurred by hauling vehicle and transporting shipments and thus economies of scale correspond to services that move vehicles that are (nearly) full. At the same time, routing shipments through this network of services and consolidation terminals increases the transportation time (and distance) over transporting them directly from origin to destination. As such, consolidation carriers must balance customer demands for shorter and shorter shipping times with their pursuit of economies of scale. The planning processes of carriers have long been assisted by solving the *Service Network Design* (*SND*) problem (Crainic, 2000), which determines the services to execute to support the timely delivery of customer shipments and the path each shipment should follow.

To execute a service, carriers need resources, including equipment and manpower. For example, while automation is becoming more and more prevalent in terminals, human resources are still used to unload shipments from inbound containers and handle and sort shipments to load into outbound containers. And resources are needed to provide the transportation itself. All modes require some type of power unit (trucks in trucking, locomotive engines in rail, planes in air, and vessels in maritime), an operator (driver, rail train operator, pilot, and captain), and sometimes a whole crew (particularly in air and sea). And of course, the vehicles and containers that hold shipments while in transit represent another resource.

These resources are often scarce, either because they are expensive to acquire (many power units), or there are few available for acquisition. The latter case is particularly true for operators. Some (Lewinski, 2014) have observed that there is or will soon be a shortage in airline pilots in the United States (primarily for air travel but likely to also impact air cargo eventually). Similarly, driver shortages have long plagued the trucking industry. While truckload carriers (those that transport customer shipments directly from their origins to their destinations) have been hardest hit by the driver shortage, often experiencing turnover rates that exceed 100%, LTL carriers have also struggled to find sufficient driver capacity. Compounding the lack of operators (pilots, drivers) today is the demographics associated with these professions; many individuals in these jobs are at or near retirement age (particularly pilots wherein retirement is mandated at 65 years of age in the United States). Similarly, many rail carriers are now reporting that they do not have (and can not acquire) enough rail cars to meet the transportation needs of their customers (Trudell, 2015; Tita, 2014). This may be particularly problematic when, e.g., intermodal rail, operators must decide every year on the number, type and home region of the vehicles they lease.

There are often rules governing the use of these resources. Power units must be periodically maintained. Operators and crews can only work (fly, drive) a certain number of hours before they must take a rest. And some rules relate to the resource's "home" location or terminal, such as a driver needing to return home periodically (such as every day or every week). Similarly, the compensation for human resources (particularly drivers), either in terms of pay rates or signing bonuses, may differ by the individual's home location. However, resources may also differ with respect to type. Thus, for example, intermodal rail cars differ significantly in the number of platforms they offer for loading containers, their length, the weight they can carry, etc.

Alternately, a carrier may wish to differentiate its vehicle fleet with respect to fuel-efficiency, and thus model as different resource types the fuel-efficient trucks for its longer hauls and the older, less fuel-efficient trucks supporting shorter services. As such, a carrier must decide not only how many resources of each type to acquire, but also where those resources should be allocated or what regions should be targeted in recruiting efforts. Of course, requiring a resource to periodically return to its home terminal also impacts the services it can support.

Carriers sometimes move a resource (particularly those that support transportation) not to support a service but to position it for the execution of a future service. Such "empty" moves are common enough for many consolidation carriers that they can represent a significant cost. As a result, much of the recent work in Service Network Design also models the need to re-position resources (sometimes called "assets"). However, much of this research assumes an unlimited number of resources and does not model that an individual resource may need to periodically return to its home terminal. More recent research (Crainic et al., 2014) models a fixed and finite set of resources, one of which is needed for each service to execute, with each resource having a home terminal to which it must return periodically. However, that research considered only one type of resource and assumed that the total number of resources (of that single type) and the assignment of each resource to a home terminal had been determined *a priori*.

In this research, we extend the work presented in Crainic et al. (2014) in multiple ways. First, we extend the model to handle multiple types of resources, each with its own set of rules dictating what it can and cannot do. We also extend the model presented there to include strategic, fleet acquisition and management-type choices: the acquisition of new resources, including their assignment to a home terminal; and the reassignment of an existing resource's home terminal. Thus, the model we present has two layers: (1) a strategic layer, wherein resource acquisition and allocation decisions are made, and, (2) a tactical layer wherein service network design decisions are made along with decisions regarding the management of resources used to execute services. This second, tactical, layer serves as an estimator to the strategic layer of the impact of its decisions on transportation costs. We also include in the tactical layer the option of using a third party's resource (at extra cost) to support the transportation associated with a service. As these are third party resources, we do not model that the carrier must return them to a home terminal; instead we assume they are available to execute the service and no further management is needed.

In addition to extending the model, we also extend the solution approach presented in Crainic et al. (2014) to solve this new model. Finally, we perform an extensive computational study to validate the use of the new model, benchmark the performance of the algorithm, and to understand how the resource-related and service network design-related components of the model interact. Fundamentally, the primary contributions of this paper are threefold: (1) the first optimization model to help consolidation carriers determine the size, mix, and allocation of their fleet with a clear picture of the impact of those decisions on their transportation costs, (2) a solution approach that can effectively solve that model, and, (3) an extensive computational study demonstrating the value of using both the model and the solution approach.

The paper is organized as follows. We begin with a detailed description of the problem in Section 2, after which we review the relevant literature to this problem in Section 3. Next, we detail the network model and mathematical formulation in Section 4. Section 5 is dedicated to the proposed matheuristic solution approach, while Section 6 presents the experimental study. We conclude in Section 7.

# 2 Problem Statement

Fundamentally, a consolidation carrier must choose a set of suppliers from which it can acquire a fleet of resources that is (potentially) comprised of resources with different capabilities (what we will call *types*). Then, with this fleet of resources, it must create a service network with which it can transport customer shipments. Together, the costs associated with acquiring this fleet and transporting customer shipments have a significant impact on carrier profitability.

As such, in this paper, we consider a problem that encompasses both strategic and tactical planning decisions for a consolidation-based carrier. While supplier selection can have a significant impact on these costs, in this paper we presume that the carrier has a pre-existing set of suppliers from which it acquires resources. We also assume the carrier transports freight through a network of terminals on services. The tactical decisions select the services to operate and determine how freight is routed through this service and terminal network, with these services and routes supported by resources which are assigned to terminals. We also focus on a problem wherein the capacity of a service and the cost of executing it can be influenced by the type of resource that executes it.

The strategic decisions determine the total number of resources of each type available and the assignment of each resource to a home terminal. Costs associated with these strategic decisions include the purchase cost of a capital asset, the additional salary and/or signing bonus associated with hiring an individual, and transportation costs associated with re-allocating a resource from one home terminal to another. These strategic decisions can have a profound impact on the cost of transporting freight. While these decisions can be (and often are) made independently, we propose to solve them jointly; in this way the strategic, resource acquisition and allocation decisions can be made with an accurate estimate of their impact on the transportation costs the carrier will 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 we refer to as the *tactical planning horizon* (e.g., six months to one year). During this period of time, the carrier must transport a set of known (forecast) customer shipments; associated with each shipment is an origin location and availability time, a destination location and due arrival time, and a size. That said, we assume shipments are divisible, and thus portions of a single shipment may follow different paths. At terminals, shipments are sorted and consolidated into vehicles (and convoys, eventually) 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 temporal characteristics: 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 to services and provide the means to perform them, the resource type dictating the service capacity. Each resource operates according to cycling routes, with the type of resource dictating the rules governing that route, supporting a sequence of services, starting and returning to its assigned terminal.

Shipments will thus be routed through the service network, 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 to start it. 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. We presume the transportation costs associated with executing a service can depend on the type of resource that executes it (e.g., vehicles with different fuel efficiencies). There are also costs associated with handling a shipment at a terminal.

As it is usual for tactical planning and service network design models, we focus on the regular and repetitive part of demand and operations. More precisely, it is assumed that shipment demands during the tactical planning horizon are representative and display a regular, repetitive pattern. Based on history, relations with customers and company operating policy, the planner selects a certain period, a week, say, to be representative of demand and operations. It is then assumed that the carrier will see the same demand repeated each week for the duration of the tactical planning horizon. Services and resource routes will therefore be selected and scheduled for the same time length, called *schedule length*. We illustrate the problem we study in Figure 1.

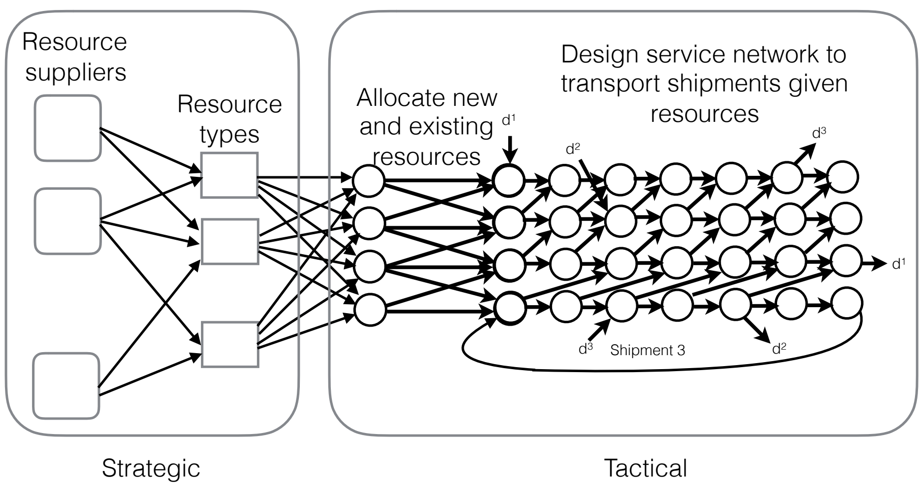



Figure 1: Illustration of problem studied

This work extends the research presented in Crainic et al. (2014), which models that a single resource is needed to execute a service, that all resources have the same capabilities (are of the same type), and that these resources are assigned to terminals to which they must periodically return. 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 resource to another, and in the extreme case a shipment may travel 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. For example, a driver may drive at most 11 hours a day after 10 consecutive hours off duty (at rest).

We consider simple sets of rules governing what movements a resource may make in this paper, namely that the resource must return to its home terminal at least once during the tactical planning horizon. However, the model and solution method we propose can be easily adapted to other cases. There are also costs, such as those incurred due to maintenance, associated with using a resource from a specific home terminal during the tactical planning horizon.

The last resource-related decision we consider is 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 incurs costs that are greater than executing the service with an owned resource.

Ultimately, the problem the planner faces is to determine the number and allocation of resources of each type that best balances the extra costs associated with resource acquisition and re-allocation with the savings these decisions enable in terms of transportation. 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 savings seen 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 a series of periods of time that equal the length of the schedule length.

# 3 Literature Review

The planning problem we study connects two types of decisions: determining the acquisition and allocation of resources and how to transport customer shipments using those resources. The resource acquisition and allocation decisions can be seen as facility location-type decisions, whereas determining how to transport customer shipments can be viewed as 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 our problem locates resources, not facilities, so we conclude this section 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 (2001a) present an optimization model that both chooses locations for (uncapacitated) facilities and designs a transportation network. This transportation network is used to route each customer shipment from its origin to the nearest available facility. The objective of the model is to minimize the combined cost of opening facilities, operating services and routing shipments. They transform the problem into an uncapacitated fixed cost network design problem that can be solved in reasonable run-times using a mixed integer programming solver. Building off this work, Melkote and Daskin (2001b) 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.

In short, and to the best of our knowledge, resource acquisition, allocation, and management decisions have not been considered in the literature on location and network design problems. As such, we next turn our attention 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 the "design-balance" constraint) 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).

Researchers (Pedersen et al., 2009) have 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., 2014) 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 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. (2009) 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. (2014) 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.

# 4 Optimization Model

We first discuss how we model the decisions made during the tactical planning horizon. During this period we assume the carrier transports shipments through a network of terminals (represented by the set $\Lambda$) on services according to a schedule of given length (to be repeated for the duration of the planning horizon). The tactical decisions determine how these services are selected and the schedule, supported by resources, is determined, as well as how shipments are routed through this network. Specifically, services are selected and scheduled over $\mathcal{T} = \{1, 2, \ldots, TMAX\}$ time periods making up the *schedule length*.

We model the operations of a carrier with a time-space network, $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where terminal activities in different periods are modeled with different nodes. Specifically, we use the node set $\mathcal{N}$ to model 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 $\mathcal{A}$ defined on $\mathcal{N}$contains two types of arcs. The first is a *service arc* and models the operation of a service between two terminals at a particular point in time. There are several possible ("parallel") service arcs, starting at different points in time, for the same pair of terminals. There are also several possible services between a same pair of terminals, those services being different in terms of capacity, speed, cost. 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 $\mathcal{A} = \mathcal{S} \cup \mathcal{H}$.

Each service $s \in S$ connects two nodes $o_s = l_t$ and $d_s = m_{t'}$. Service $s$ starts at $t$ from terminal $l$ and ends at terminal $m$ at time $t'$. When the cost of executing service $s$ is independent of the resource executing it, we model the cost with the parameter $f_s$. Similarly, we model the capacity of service $s$ with the parameter $u_s$ when it does not depend on the resource that executes it. Later in this section we will discuss how we model cases where these parameters depend on the resource that executes the service. 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 we assume these arcs are 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 from terminal $l$ available in period $t$ that needs to be delivered at terminal $m$ during period $t'$ as a commodity with index $k$, origin node $o(k) = l_t$, and destination node $d(k) = m_{t'}$. We denote the size of this shipment as $w_k$. The set of all shipments is represented by $\mathcal{K}$. The model prescribes the route from $o(k)$ to $d(k)$ for commodity $k$ through the continuous and non-negative variables $x_a^k$. Such variables are defined for each $a \in \mathcal{A}$ and represent the portion of demand associated with commodity $k$ that travels on arc $a$, which may be a service or a holding arc. To model the flow of a commodity from its source to its sink, we let the notation $d_i^k$, represent the net in/outflow of commodity $k$ that is required at node $i$ in the time-space network. As such, we have that $d_i^k = w_k$ when $i = o(k)$, $d_i^k = -w_k$ when $i = d(k)$, and $d_i^k = 0$ otherwise. We associate the variable cost $c_a^k$ with commodity $k$ traveling on arc $a \in \mathcal{A}$. For a service arc, this parameter can model handling costs associated with loading the shipment into a container 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.

As discussed, we presume that the operation of a service requires a resource. Resources may be of different types, and must follow routes which we model with *cycles*. As such, we let the set $\mathcal{R}$ define the types of resources available for operation. We then define $\theta_l^r$ as the set of potential cycles a resource of type $r$ that is assigned to terminal $l$ can execute. Then, we define $\theta^r = \cup_{l \in \Lambda} \theta_l^r$ and $\theta = \cup_{r \in \mathcal{R}} \theta^r$. To link services and resources, for each service $s \in \mathcal{S}$ and cycle $\tau \in \theta$, we define a 0-1 parameter $q_s^\tau$ that indicates whether service $s$ is executed by a resource that follows cycle $\tau$. Regarding costs, we model the cost of operating a resource of type $r$ that is assigned to terminal $l$ with the parameter $F_l^r$. This parameter is independent of the cycle operated, and thus can be used to model fixed costs such as salaries or maintenance. When the cost and capacity of service $s$ depend on the type of resource executing the service, we model those values with the parameters $f_s^r$ (cost) and $u_s^r$ (capacity). Finally, we define $z_\tau^r$ as a 0-1 variable indicating whether cycle $\tau \in \theta^r$ is selected for a resource of type $r \in \mathcal{R}$.

For each service $s$, we also model the option of executing that service, albeit with the use of a third party-owned resource of type $r \in \mathcal{R}$. We model this choice with the binary variable $y_s^{re}$ and associate with it the fixed cost $f_s^{re}$. 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 $f_s^r$, but that the transportation cost associated with executing the service with a third party carrier will be higher.

We next turn our attention to modeling the strategic decisions the planner must make. Conceptually (and as we will see later, algorithmically), we model these choices, in conjunction with the tactical planning decisions discussed above, in the manner depicted in Figure 2. To simplify the presentation, in this figure we illustrate the decisions related to a single resource type; others are modeled similarly. Fundamentally, we add to the time-space network on which service choice and commodity transportation decisions are modeled, a layer that represents the choice of resource acquisition and allocation decisions. There are two types of nodes in this layer. The first is an *Acquisition node*, which we denote by $A$, representing the acquisition of a new resource. We connect this node 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 is used to model 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 assigned to terminal $l$ to terminal $l'$. We let $\Lambda^+$ denote the set of terminals $\Lambda$ along with the Acquisition node.

We model both types of decisions with the variable $h^r_{ll'}$ with which we associate the cost parameter $H^r_{ll'}$. When $l$ corresponds to the Acquisition node, the variable represents the purchase of a new resource of type $r$ and its subsequent allocation to terminal $l'$. As such, if the resource being modeled is equipment, $H^r_{ll'}$ could model the acquisition cost, only amortized. If the resource is an individual, this parameter could model wages during the planning horizon considered by the model and some amortization of a signing bonus paid to the individual. When $l$ represents an existing terminal then the variable $h^r_{ll'}$ corresponds to the allocation of a resource of type $r$ that is currently assigned to terminal $l$ to terminal $l'$. When $l \neq l'$, this variable represents re-allocation and $H^r_{ll'}$ represents any costs associated with such an action, such as transportation. When $l = l'$ this variable represents leaving resources at their currently assigned terminal, in which case we anticipate $H^r_{ll'} = 0$. Finally, as we consider cases wherein a carrier already has a fleet in place, we let the parameter $I^r_l$ represent the number of resources of type $r$ that are initially assigned to terminal $l$.

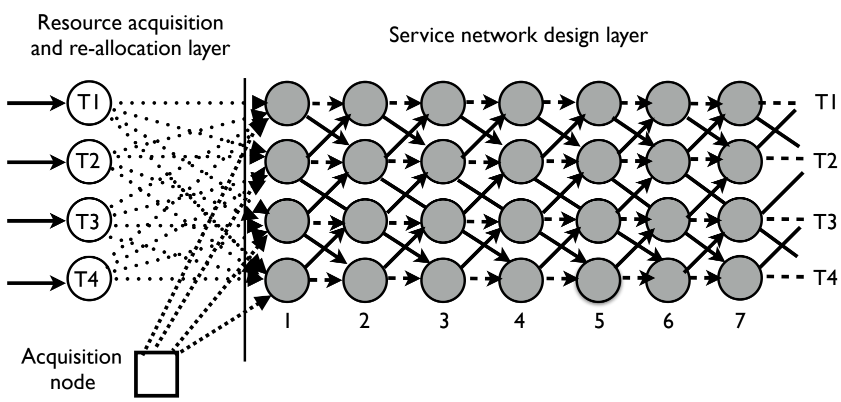



Figure 2: Network model of strategic and tactical decisions

Ultimately, we seek to solve what we call the *Scheduled Service Network Design with Resource Acquisition and Management* (*SSND-RAM*) problem, which seeks to

$$\text{minimize} \sum_{r \in \mathcal{R}} \sum_{l,l' \in \Lambda^+} h^r_{ll'} H^r_{ll'} + \sum_{r \in \mathcal{R}} \sum_{l \in \Lambda} F^r_l \sum_{\tau \in \theta^r_l} z^r_\tau + \sum_{r \in \mathcal{R}} \sum_{s \in S} f^r_s \sum_{\tau \in \theta^r} q^\tau_s z^\tau_r + \sum_{s \in S} \sum_{r \in \mathcal{R}} y^{re}_s f^{re}_s + \sum_{k \in \mathcal{K}} \sum_{a \in \mathcal{A}} x^k_a c^k_a \quad (1)$$

subject to

$$\sum_{l' \in \Lambda} h^r_{ll'} = I^r_l, \forall r \in \mathcal{R}, l \in \Lambda^+, \tag{2}$$

$$\sum_{\tau \in \theta^r_{l'}} z^r_\tau \leq \sum_{l \in \Lambda^+} h^r_{ll'}, \forall r \in \mathcal{R}, \forall l' \in \Lambda \tag{3}$$

$$\sum_{a \in \mathcal{N}^+(i)} x^k_a - \sum_{a \in \mathcal{N}^-(i)} x^k_a = d^k_i, \forall i \in \mathcal{N}, k \in \mathcal{K} \tag{4}$$

$$\sum_{k \in \mathcal{K}} x^k_s \leq \sum_{r \in \mathcal{R}} u^r_s \left( \sum_{\tau \in \theta^r} q^\tau_s z^r_\tau + y^{re}_s \right), \forall s \in S \tag{5}$$

$$\sum_{r \in \mathcal{R}} \left( \sum_{\tau \in \theta^r} r^\tau_s z^r_\tau + y^{re}_s \right) \leq 1, \forall s \in S \tag{6}$$

$$h^r_{ll'} \in \mathbb{Z}^+, \forall r \in \mathcal{R}, l, l' \in \Lambda^+ \tag{7}$$

$$x^k_a \geq 0, \quad \forall a \in \mathcal{A}, k \in \mathcal{K}, \tag{8}$$

$$z^r_\tau \in \{0,1\}, \forall r \in \mathcal{R}, \tau \in \theta^r \tag{9}$$

$$y^{re}_s \in \{0,1\}, \forall r \in R, s \in S \tag{10}$$

The objective is to minimize the total cost of resource acquisition and allocation decisions and the decisions made to support transporting customer shipments. The first term in the objective models the cost of acquiring and re-allocating resources, while the second models fixed costs associated with putting a resource into use (irrespective of how they are used). The third term models the transportation costs incurred by owned resources, while the fourth models transportation costs incurred to secure third-party resources. Finally, the last term models shipment-level routing costs.

Constraints (2) ensure that all resources of type $r$ that are initially allocated to terminal $l$ are either left at $l$ or re-allocated. Constraints (3) link the strategic resource acquisition and allocation decisions that determine the number of resources available at terminal $l'$ with the tactical decision of how many resources from $l'$ are to be used to execute services. Note the summation over $\Lambda^+$ in Constraint (3) enables the use of resources that are newly acquired. Constraints (4), (5), and (6) are variants on constraints often seen in Service Network Design models. Constraints (4) ensure that each commodity is routed from its origin node to its destination node. Constraints (5) ensure that commodities only travel on a service when it is supported by a resource (either carrier or externally-owned) and that the capacity for that service is not exceeded. Constraints (6) ensure that each service is executed at most once. Finally, Constraints (7), (8), (9), and (10) define the domains of the variables in this problem. There are other, valid, constraints for this problem, such as disaggregated versions of Constraint (5). However, as these are not the focus of this research we omit them from this model.

# 5 Solution Approach

There are two challenges associated with producing a high-quality solution to the SSND-RAM. First, even for moderately-sized instances, the set of resource cycles, $\theta$, is too large to be enumerated in a reasonable run-time. Thus, an effective solution procedure must first determine a set of cycles that are likely to appear in a high-quality solution. And, when making this determination, the solution approach must also determine the appropriate source for each cycle. Second, given a set of cycles, the solution approach must extract a solution that satisfies the constraints of the problem and is of high quality.

The solution approach we propose overcomes both of these challenges by adapting and extending techniques that were developed by Crainic et al. (2014) as part of a solution approach for the SNDRC, the tactical planning portion of the SSND-RAM, wherein there are a fixed number of resources assigned to each terminal. Regarding the first challenge, the solution procedure we propose uses column generation (Barnhart et al., 1998; Desaulniers et al., 2005) to determine the set of resource cycles that can be included in a solution. Of course column generation is an algorithm for solving linear programs. As such, and regarding the second challenge, to produce a high-quality solution to the SSND-RAM (an integer program), the approach we propose uses steps from both Slope Scaling (Crainic et al., 2004; Kim and Pardalos, 1999) and Matheuristics (Archetti et al., 2008; Chouman and Crainic, 2015; Vu et al., 2013; De Franceschi et al., 2006; Hewitt et al., 2010).

More specifically, the solution approach presented in Crainic et al. (2014) dynamically generates, via column generation, resource cycles in order to design and support a service network to enable the flow of commodity demands. As the optimization model studied in Crainic et al. (2014) focuses on a situation wherein the fleet of resources is fixed, that approach recognizes limits on the number of resource cycles that can originate at each terminal. The approach explicitly recognizes those limits by solving instances of the SNDRC model, albeit with a limited set of cycles, to produce primal solutions.

The optimization model studied in this paper differs significantly from the one solved by the algorithm presented in Crainic et al. (2014). As mentioned previously, it differs in three fundamental ways: (1) the source of each resource cycle, whether it is acquired or re-allocated, must be determined, (2) different types of resources, with different types having different capabilities and cost structures, are considered, and, (3) the option to outsource the execution of individual services is considered. In this section, we will discuss at a high level how we have adapted the column generation and slope scaling portions of the algorithm presented in Crainic et al. (2014) to handle each of these model differences, leaving technical details to the Appendix. However, fundamentally, the main steps in the algorithm can be seen in the following flow chart (Figure 3):

## 5.1 Adapting the column generation procedure

The algorithm presented in Crainic et al. (2014) dynamically generates resource cycle variables with a given origin terminal via column generation. In general, column gen-

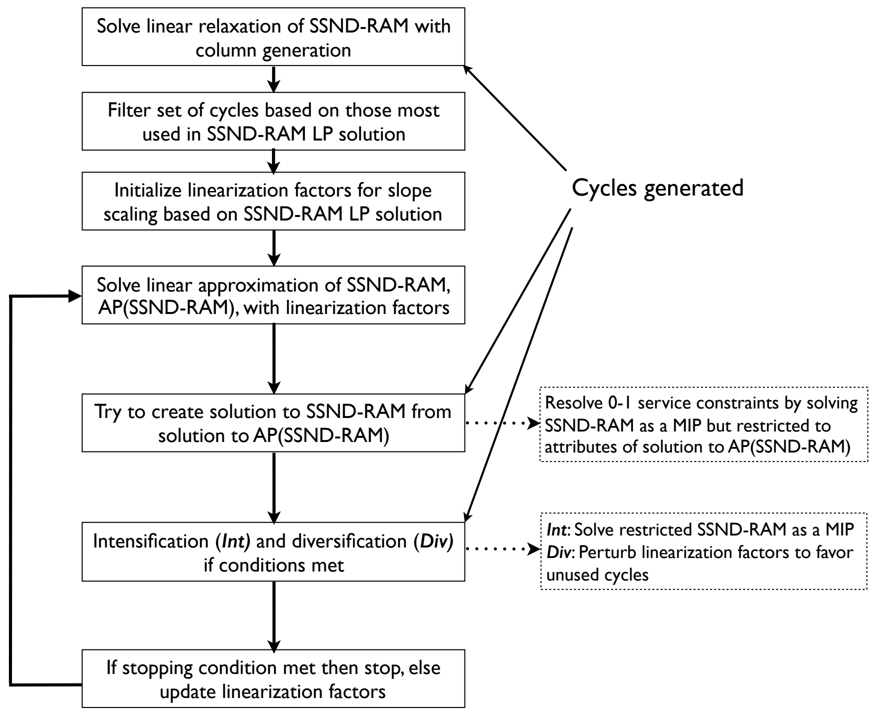



Figure 3: Steps of solution approach

eration procedures perform two steps: (1) the solution of a restricted master problem (a linear program), and, (2) a pricing procedure that generates new variables that have negative reduced costs (and thus appear as promising candidates for appearing in the optimal solution to that linear program). To solve the SSND-RAM, we adapted both steps of the column generation portion of the procedure presented in Crainic et al. (2014) for solving the SNDRC. Regarding the first step, the restricted master solved at an iteration is changed from an instance of the SNDRC to an instance of the SSND-RAM. In this way, our column generation procedure explicitly recognizes the opportunity to acquire new resources and reposition existing ones, the presence of multiple types of resources, and the option of outsourcing the execution of a service.

We also adapted the pricing procedure. While the structure of the pricing problem is the same (i.e., it still seeks to find a least-cost cycle in an acyclic graph that satisfies some side constraints), the objective is different. First, the presence of Constraints (3) in the SSND-RAM yields dual variables that reflect the opportunity to acquire new or reposition existing resources, which then appear in the reduced cost formula for a cycle variable. Secondly, the reduced cost formula for a cycle variable also reflects the fact that the cost of executing a service, as well as its capacity, can depend on the resource that executes it (e.g., through parameters $f^r_s, u^r_s$). We refer the reader to Equation (11) to see the explicit formula for the reduced cost of cycle variables in the SSND-RAM.

Finally, the presence of multiple types of resources necessitates further adapting the column generation procedure. Fundamentally, the algorithm proposed in this paper generates cycles that represent the use of resources of different types by solving different pricing problems, one for each type of resource, in the column generation procedure just described. As just discussed, that multiple types of resources are considered by the SSND-RAM impacts the objective of the pricing problem that is solved for each type of resource. Similarly, when different types of resources must observe different rules in their itineraries (e.g., one type of resource must return home every day whereas another type only needs return home once a week), such differences can be encoded in the constraint set of the respective pricing problem.

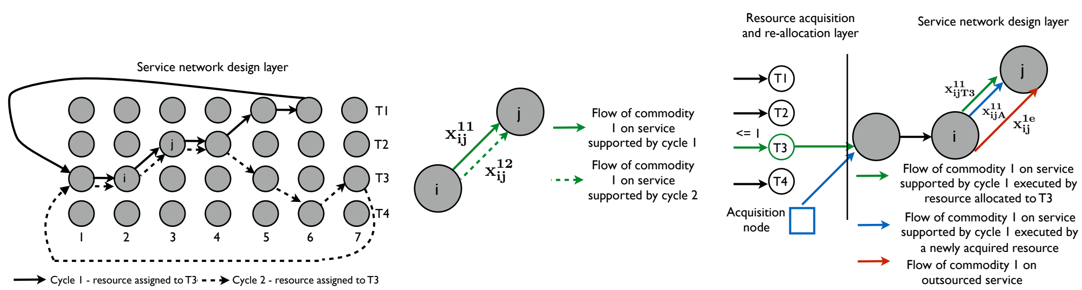
## 5.2 Adapting the slope scaling procedure

When using slope scaling (Kim and Pardalos, 1999; Crainic et al., 2004) to produce solutions to an integer program, an approximation of the problem is repeatedly solved with an objective function that is parameterized by *linearization factors*. Each time the approximation problem is solved, its solution is used to derive a solution to the integer program, and the linearization factors are updated so that the cost of the two solutions agree. Typically, to produce solutions to an integer program (such as the SSND-RAM) with slope scaling, the approximation problem is the linear relaxation of the original integer program. Our approximation problem, which we call AP(SSND-RAM), is formed by both relaxing the integrality constraints and removing other constraints, leaving what is essentially a multi-commodity flow problem. The approximation problem we define is similar to what was presented in Crainic et al. (2014) for the SNDRC, which linearized cycle variables $z_\tau$ (note the SNDRC only considered one type of resource). However, in the SSND-RAM we must also linearize the acquisition and allocation variables, $h^r_{ll'}$, and the outsourcing variables $y^{re}_s$. We first discuss how we adapt the slope scaling procedure when there is just a single type of resource that can be used.

Specifically, the approximation problem we solved for the SNDRC involved flow variables that represented the amount of commodity $k$ that was transported on a service supported by a resource following a given cycle $\tau$. The purpose of the linearization factor for such a flow variable is to capture the cost of both the execution of the service and the resource cycle that supports that service. We illustrate in Figure 4(a) and 4(b) the slope scaling approach presented in Crainic et al. (2014). In Figure 4(a), we have two cycles, both of which represent the itinerary of a resource assigned to terminal T3. Focusing on the service $(i, j)$ (which is contained in each cycle), and assuming there are two commodities ($k = 1, 2$), we have two flow variables in the approximation problem, one for each cycle (illustrated in Figure 4(b)).

The slope scaling approach for the SSND-RAM differs in what the flow variables (and hence the linearization factors) in the approximation problem represent. As we must also linearize acquisition and allocation variables, as well as outsourcing variables, we have two types of flow variables in the AP(SSND-RAM). The first type of flow variable represents the flow of commodity $k$ on a service supported by an owned resource following a given cycle $\tau$. To derive these flow variables, we first decompose a cycle variable that represents an itinerary for a resource into multiple variables, with each representing that same itinerary, but operated with resources from different sources. In a sense, there are multiple copies of the flow variables seen in the approximation model for the SNDRC, with a copy for each potential source of that owned resource (e.g., new or re-allocated).

For example, in Figure 4(c), flow variable $x_{ijT3}^{11}$ represents the flow of commodity 1 on service $(i, j)$ on a resource that was initially allocated to terminal T3 and is following the itinerary modeled by cycle 1. For this variable, the linearization factor will include estimates of the costs of executing the service and those associated with the cycle the resource is following (note the resource did not need to be re-assigned). Conversely, the flow variable $x_{ijA}^{11}$ is similar, albeit models the use of a resource that is newly acquired. For this flow variable, the linearization factor also includes an estimate of the cost of acquiring the resource. The second type of flow variable represents the flow of commodity $k$ on a service whose execution is outsourced (and is illustrated in Figure 4(c) with variable $x_{ij}^{1e}$). For this variable, the linearization factor will include an estimate of the cost of outsourcing the service. Note that while not illustrated in Figure 4(c), we would have similar flow variables for cycle 2. Finally, whereas the SNDRC assumes a single type of resource, the SSND-RAM allows for multiple types of resources. To accommodate this in the slope scaling procedure, we create copies of each flow variable, one for each type. For example, if there were two types of resources available, we would create two copies of $x_{ijT3}^{11}$, with each representing a different type of resource following the itinerary modeled by cycle 1.




(a) Cycles for a resource assigned to T3 (b) Auxiliary flow variables in Crainic et al. (2014) (c) Auxiliary flow variables in proposed algorithm

Figure 4: Adapting cycle generation approach for SNDRC to SSND-RAM

Our slope scaling procedure repeatedly solves an approximation of the SSND-RAM that is realized by both linearizing integer variables and removing other constraints. As such, rounding a solution to the AP(SSND-RAM) need not yield a feasible solution to the SSND-RAM. Thus, to create a feasible solution to the SSND-RAM, the approach examines a solution to the AP(SSND-RAM) to determine which cycles appear as if they could appear in a good solution. With this set of cycles $\tilde{\theta}$, an instance of the SSND-RAM is solved (albeit restricted to the use of those cycles), which will in turn yield a feasible solution to the SSND-RAM. Further details regarding the algorithm can be found in the Appendix.

# 6 Computational Results

We next report on an extensive computational study. First, we validate the use of the SSND-RAM. Specifically, we compare the resource acquisition and allocation and transportation decisions prescribed by the model proposed in this paper, with plans prescribed by other techniques. We also study how jointly making these strategic and tactical decisions impacts performance metrics associated with the plans prescribed. Second, having shown the value in using the SSND-RAM, we study (computationally) the effectiveness of the solution approach proposed, which we refer to as *SSCG(SSND-RAM)*, *Slope Scaling with Column Generation for the the Scheduled Service Network Design with Resource Acquisition and Management Problem.* All experiments were performed on a cluster of computers that have 2 Intel Xeon 2.6 GHz processors and 48 GB of RAM and were running Scientific Linux 6.1. We used CPLEX 12.4 to solve linear and mixed integer programs. We next describe the instances used in the computational study.

## 6.1 Problem instances and parameters

We perform experiments on three sets of instances. Each set is based on 35 instances from the literature (Crainic et al., 2014; Andersen et al., 2011) that were inspired by a rail-based case study. These 35 instances can be broken down into 7 classes, each of which contains 5 instances. As these instances form our first set, we present details regarding their sizes along multiple dimensions in Table 1; we label these instance classes 6-12 to maintain consistency with Andersen et al. (2011). Note that these instances have a single type of resource and thus each service has a single, potential, capacity.

Regarding the cost structure of these instances, the cost associated with executing a service with an owned resource, $f_s^r$, is set to 50 times the length of the service. The analogous cost associated when using a third-party resource, $f_s^{re}$, is $m \times f_s^r$ with $m = 4$ in order to balance the use of owned resources and outsourcing. The cost associated with acquiring a new resource ($H^r_{Al'}, l' \in \Lambda$) is 1,000. The cost associated with repositioning a resource ($H^r_{ll'}, l \in \Lambda, l' \in \Lambda$) is generated randomly using the function (rand()%5+1)∗100 (rand()%5 generates a random number in the interval [0..4]. To understand the sensitivity of the effectiveness of SSCG(SSND-RAM) to the cost ratio own/outsource, we consider various values for the fixed cost $F_l^r$ associated with using a resource; in some experiments we choose values that favor the use of owned resources and in others we choose values that favor outsourcing. We refer the reader to the original source of these instances (Andersen et al., 2011) for details on how other instance parameter values were derived.

We also seek instances of the SSND-RAM that can be solved with an off-the-shelf MIP solver in a reasonable run-time in order to benchmark the performance of SSCG(SSND-RAM). However, the instances in Set 1 are too large to do so; in fact the number of potential cycles is too large for the set to be enumerated in a reasonable run-time. As such, from the 35 instances in Set 1, we derive 14 smaller instances by truncating the planning horizons of the original instances. We present details regarding the instances in this set (Set 2) in Table 2. Note that the instances in Set 2 also have a single type of

| Instance class | Terminals | Services | Periods | Service+Holding arcs | Commodities |
|---|---|---|---|---|---|
| 6 | 5 | 15 | 40 | 600+200 | 200 |
| 7 | 5 | 15 | 50 | 750+250 | 400 |
| 8 | 7 | 30 | 30 | 900+210 | 200 |
| 9 | 7 | 30 | 30 | 900+210 | 400 |
| 10 | 7 | 30 | 50 | 1500+350 | 300 |
| 11 | 10 | 40 | 30 | 1200+300 | 200 |
| 12 | 10 | 50 | 30 | 1500+350 | 100 |

Table 1: Set 1: Rail-based instances adapted from Andersen et al. (2011).

resource.

| Instances | Terminals | Services | Commodities | Service + Holding Arcs | |
|---|---|---|---|---|---|
| | | | | 20 periods | 25 periods |
| 1-4 | 5 | 15 | 200 | 300+100 | 375+125 |
| 5-6 | 7 | 30 | 400 | 600+140 | 750+175 |
| 7-8 | 7 | 30 | 200 | 600+140 | 750+175 |
| 9-10 | 7 | 30 | 300 | 600+140 | 750+175 |
| 11-12 | 10 | 40 | 200 | 800+200 | 900+250 |
| 13-14 | 10 | 50 | 100 | 1000+200 | 1250+250 |

Table 2: Set 2: Small instances used for calibration and benchmarking against MIP solver.

Finally, to validate the use of the SSND-RAM, we seek to compare solutions to the SSND-RAM that are of provably high-quality, against provably high-quality solutions from other optimization-based planning approaches. As such, we derive a third set of eight instances from Set 2, but add a second type of resource. To derive these instances, we draw on data from the American Transport Research Institute[1] to determine capacity and cost data for two types of resources that mimic the use of large (Resource type 1) and small (Resource type 2) trucks. Details regarding these instances can be found in Table 3. With these instances, we model that the smaller resources may be limited to shorter, regional moves, by limiting the services they can operate to the shortest (in terms of length) 80% of all possible services. We also model that smaller resources may incur less overhead time-wise (e.g., with less capacity, less time is spent loading/unloading freight). To do so, we presume that some services require one less period to execute when supported by a small resource (type 2) than by a large resource (type 1). Similarly, as fuel efficiency can depend on vehicle size, we presume that the cost of executing a service is sometimes cheaper when supported by a resource of type 2. Finally, we note that we use a single period in these instances to model three hours of transit.

[1]www.atri-online.org/wp-content/uploads/2014/09/ATRI-Operational-Costs-of-Trucking-2014-FINAL.pdf

| Instance | #Terminal | #Commodity | Resources type 1 | | Resources type 2 | |
|---|---|---|---|---|---|---|
| | | | #Service | Capacity | #Service | Capacity |
| 1 | 5 | 200 | 15 | 50 | 10 | 30 |
| 2 | 5 | 200 | 15 | 50 | 10 | 30 |
| 3 | 7 | 200 | 30 | 50 | 24 | 30 |
| 4 | 7 | 200 | 30 | 50 | 24 | 30 |
| 5 | 7 | 300 | 30 | 50 | 24 | 30 |
| 6 | 7 | 300 | 30 | 50 | 24 | 30 |
| 7 | 10 | 100 | 50 | 50 | 40 | 30 |
| 8 | 10 | 100 | 50 | 50 | 40 | 30 |

Table 3: Set 3: Instances for testing impact of multiple resources.

With this last set of instances, we begin our computational analysis by validating the use of the SSND-RAM in a planning process.

## 6.2 Validating the SSND-RAM

The SSND-RAM differs from existing planning models for consolidation carriers in multiple ways. Most notably, it jointly makes strategic, resource acquisition and allocation decisions and tactical, service network design decisions. By doing so, the service network design component of the SSND-RAM provides an accurate estimate of the impact of the strategic decisions. However, one need not make decisions in this manner. As such, in this section, we compare solutions to the SSND-RAM with other methods for making strategic, resource-related acquisition and allocation decisions. To do so, we perform our experiments on instances from Set 3, nearly all of which we are able to solve to optimality. We also, for each instance in Set 3, derive three more instances with respect to different levels of demand, which we model with the number of commodities in the instance. For example, for our first experiment, we create from each instance in Set 3, three new instances, each with a differently-sized commodity set ($|\mathcal{K}| = 25$,50, and 100).

All experiments reported on in this section are the result of solving MIPs with CPLEX 12.4 to an optimality tolerance of $10^{-4}$ and with a 15 hour time limit. In this way, when validating the SSND-RAM, we are doing so based upon a comparison of high-quality solutions. When comparing solutions, we primarily focus on the relative gap in their total costs, calculated as $\frac{val(SSND-RAM)-val(Alt-method)}{val(SSND-RAM)}$, where *Alt-method* refers to the method we are comparing against. From this statistic, as it is based on primal solutions, we estimate an upper bound on the cost difference associated with executing the plan prescribed by the solution to the SSND-RAM with the plan prescribed by the alternative method. That said, as we are not always able to compare solutions that are optimal, we often calculate the gap in the total cost of the SSND-RAM and a dual bound on the total cost of the alternative method, or, $\frac{val(SSND-RAM)-val\_dual(Alt-Method)}{val(SSND-RAM)}$. From this statistic we have a lower bound on the potential cost difference. Finally, we note that as many of these alternative methods involve solving the SNDRC, albeit with multiple types of resources, we adapted the model presented in Crainic et al. (2014) to handle multiple types of resources.

While the SNDRC (Crainic et al., 2014) modeled the need for a resource to support the execution of a service, it did not model the opportunity to instead outsource the execution of a service to a third-party carrier. As such, we first focus on the value in modeling this opportunity. To do so, we constructed instances from those in Set 3 where there are enough resources of each type at each terminal that acquisition and re-allocation decisions need not be modeled. In these experiments, we then constructed and solved an instance of the SNDRC, wherein outsourcing is not modeled, and an instance of the SSND-RAM, wherein it is. We then compared the objective function values (and dual bounds on those values) for each solution.

We report the results of these experiments in Table 4. Column "Opt. gap" reports the optimality gap reported by CPLEX when solving each model. Column "Utilization" reports the percentage of owned capacity that is used to transport flow (Equation 26 in the Appendix presents the formula we use to calculate utilization). Column "Empty move" reports on the percentage of movements made by owned resources that do not carry flow. The last two columns of Table 4 report our upper and lower bounds on the potential reduction in costs of the plan prescribed by the solution to the SSND-RAM. As noted, the upper bound is calculated as $\frac{val(SSND-RAM)-val(SNDRC)}{val(SSND-RAM)}$, where each $val$ corresponds to the objective function value of the best primal solution produced by CPLEX. Similarly, the lower bound is calculated as $\frac{val(SSND-RAM)-val\_dual(SNDRC)}{val(SSND-RAM)}$, wherein $val\_dual$ is the dual bound reported by CPLEX on the instance of the SNDRC. Each row corresponds to an average over the eight instances in Set 3, and a given number of commodities.

| | SNDRC | | | SSND-RAM | | | Cost difference estimates | |
|---|---|---|---|---|---|---|---|---|
| $\lvert\mathcal{K}\rvert$ | Opt. gap | Utilization | Empty move | Opt gap | Utilization | Empty move | Upper bound | Lower bound |
| 25 | 0.22% | 38.74% | 8.51% | 0.18% | 44.65% | 0.49% | -9.34% | -9.12% |
| 50 | 0.31% | 46.39% | 5.87% | 0.28% | 50.85% | 0.00% | -5.86% | -5.54% |
| 100 | 1.29% | 57.64% | 3.01% | 0.75% | 62.70% | 0.00% | -3.15% | -1.82% |
| All | 2.22% | 65.15% | 1.35% | 1.81% | 69.21% | 0.18% | -2.36% | -0.10% |
| Average | 1.01% | 51.98% | 4.69% | 0.76% | 56.85% | 0.17% | -5.18% | -4.15% |

Table 4: The benefit of modeling outsourcing decisions in the SSND-RAM

We see in Table 4 that the opportunity to outsource the execution of a service brings great savings potential. While the small optimality gaps for the respective solutions indicate that the upper bounds on the savings potential are likely exact, the lower bounds also indicate great savings potential (over 4% on average). From the "Utilization" and "Empty move" columns we conclude that these savings are likely achieved by outsourcing the execution of services that would otherwise carry little flow, or would potentially force a resource to make an empty move.

Having established the savings potential associated with explicitly modeling outsourcing, we next study the savings potential associated with the SSND-RAM making resource acquisition and allocation decisions. To do so, we compare the costs of high-quality solutions to the SSND-RAM with two solutions produced by two alternative strategies of resource assignment. The first method is a shipment volume-based heuristic that assigns a fleet of resources of each type of fixed size to terminals, with the number assigned to a terminal proportional to the total volume of shipments originating at that terminal. With this assignment of resources, the tactical transportation planning problem is an instance of the SNDRC (Crainic et al., 2014). We call this overall strategy "POSITION + SNDRC," ("P+S" in Table 5). The second alternative method we evaluate is one that lets the tactical, transportation layer dictate the allocation of resources. Specifically, while we assume a set of resources assigned to each terminal, we initially ignore these bounds and solve an instance of the SNDRC wherein there are (effectively) an unlimited number of resources at each terminal. This problem yields, in a sense, the "optimal" allocation of resources of each type from the service network design perspective. We then, given the actual allocation of resources to terminals, solve an assignment problem to reposition and acquire new resources to meet this optimal allocation. We call this overall strategy "SNDRC + POSITION" ("S+P" in Table 5).

Thus, the "POSITION+SNDRC" method requires a total number of resources of each type that are then allocated to terminals via the volume-based heuristic. Also, the SSND-RAM requires an initial number of resources of each type at each terminal (parameter $I_l^r$). To derive this information, we examine the solutions to the SNDRC that are the basis of the results in Table 4. Specifically, we examine the number of resources of each type that are used in those solutions to the SNDRC to derive the parameters $I_l^r$ for the SSND-RAM. For "POSITION+SNDRC," we calculate the total number of resources of each type that are used in those solutions to the SNDRC to determine the number of resources of each type that will then be allocated via the volume-based heuristic. To compare solutions, we use the resource allocations from both the "POSITION+SNDRC" and "SNDRC+POSITION" methods to derive solutions to the SSND-RAM, and calculate the resulting objective function value. In these experiments, the dual bound used to derive the lower bound on the savings estimate is from the SNDRC portion of the alternative method.

| | Opt. gap | | | Upper bound on cost difference | | Lower bound on cost difference | |
|---|---|---|---|---|---|---|---|
| $\lvert\mathcal{K}\rvert$ | S+P | P+S | SSND-RAM | SSND-RAM vs P+S | SSND-RAM vs S+P | SSND-RAM vs P+S | SSND-RAM vs S+P |
| 50 | 0.28% | 0.75% | 0.25% | -10.81% | -7.99% | -5.88% | -4.96% |
| 100 | 1.04% | 1.85% | 1.19% | -5.35% | -4.15% | -3.63% | -1.40% |
| Average | 0.66% | 1.30% | 0.72% | -8.08% | -6.07% | -4.76% | -3.18% |

Table 5: The benefit of modeling resource acquisition and allocation decisions in the SSND-RAM

We report in Table 5 the optimality gaps of the MIPs solved with each method and lower and upper bounds on the savings estimate for instances with 50 and 100 commodities. The optimality gap columns indicate that we are looking at high quality solutions from each method. The savings columns indicate that the SSND-RAM produces plans that have significant potential for savings; the smallest lower bound estimate is still 1.40%. We continue our comparison in Table 6 where we compare the utilization of services executed by owned resources and the percentage of empty moves in the solutions from each method. We see that integrating the resource acquisition and allocation decisions with the tactical transportation planning decisions leads to higher utilization and fewer empty moves.

Finally, we focus on the ability of the SSND-RAM to effectively manage capacity. To

| #Commodity | SNDRC+POSITION | | POSITION+SNDRC | | SSND-RAM | |
|---|---|---|---|---|---|---|
| | Utilization | Empty move | Utilization | Empty move | Utilization | Empty move |
| 50 | 46.41% | 9.30% | 46.41% | 10.58% | 50.32% | 0.00% |
| 100 | 57.52% | 3.63% | 44.39% | 5.29% | 62.98% | 0.14% |

Table 6: Average utilization and empty move %

do so, we conduct an experiment wherein we increase the capacity and acquisition cost of resources of type 2, by steps (20%, 40%, 60%, and 80%). We then solve the resulting SSND-RAM and report the number of resources that are acquired relative to the number acquired with the initial level of capacity of type 2 resources. We also report the decrease in total cost, relative to the solution produced with the initial level of capacity of type 2 resources. We report the results in Table 7, wherein we report the relative number of resources, as well as the savings. We see that the SSND-RAM is able to take advantage of the capacity that it has available, as the greater the capacity of type 2 resources, the lesser the number of resources needed and more potential savings.

| | Increase in capacity and cost of resource type 2 | | | |
|---|---|---|---|---|
| vs Baseline | +20% | +40% | +60% | +80% |
| Total resources | 92.97% | 93.10% | 87.86% | 87.86% |
| Cost difference | -2.63% | -4.65% | -6.06% | -6.09% |

Table 7: Validating how SSND-RAM manages capacity

To conclude, we note that we have run other experiments wherein the rules governing the itineraries that resources of each type must follow change. We again saw the SSND-RAM shift its usage of resources to those that had the less constraining rules. Ultimately, we conclude that the SSND-RAM produces effective plans, and that it should be solved as part of a process of determining fleet mix, sizing, and allocation decisions.

## 6.3 Benchmarking the SSCG(SSND-RAM)

Having validated the use of SSND-RAM by studying solutions on instances that are small enough to be solved with an off-the-shelf solver, we next turn our attention to benchmarking the computational performance of a calibrated version of SSCG(SSND-RAM); see Section 8.5 for details regarding calibration. We benchmark SSCG(SSND-RAM) against two other methods. First, to derive an absolute estimate of the ability of SSCG(SSND-RAM) to produce high-quality solutions, we benchmark its performance against a commercial MIP solver (CPLEX 12.4) on instance Set 2. These instances are small enough (because the planning horizon is truncated) that we are able to enumerate the set of cycles, $\theta$, in a reasonable amount of time. Second, to derive a relative estimate of the ability of SSCG(SSND-RAM) to produce high-quality solutions on larger instances, we benchmark the solutions it produces for instances in Set 1 against those produced by an alternate, column generation-based, heuristic.

For the first set of experiments (on the instances in Set 2), CPLEX was executed with an optimality tolerance of 1% and a time limit of either one or ten hours. We considered two planning horizon lengths; 20 and 25 periods. For the 20-period instances, CPLEX was executed for one hour and we report in Table 8 the optimality gap associated with the primal solution produced and the dual bound derived by CPLEX at termination. For the 25-period instances we executed CPLEX for ten hours and report the resulting optimality gap after one and ten hours. Numbers that appear in parentheses indicate the number of instances wherein CPLEX was not able to find a primal solution in the time allotted. We also report the optimality gap associated with the primal solution produced by SSCG(SSND-RAM) after one hour as measured against the dual bound produced by CPLEX at termination (one hour for 20-period instances and ten hours for 25-period instances).

In these experiments we consider three values for the cost, $F_l^r$, of using a resource assigned to terminal $l$; 500, 1000, and 2000. We note that we assume this parameter does not vary with respect to the terminal (e.g., $F_l^r = F_{l'}^r, \quad \forall l, l' \in \Lambda, r \in \mathcal{R}$). We also consider two sizes, *small* and *large*, for the number of owned resources. We estimated the *small* size in such a way that we believed either resources would need to be acquired or externally owned resources would need to be used to support executed services. We estimated the *large* level in such a way that we believed owned resources would be enough to support the services executed (although repositioning might still be advisable).

| | Size of resource fleet | | | | | |
|---|---|---|---|---|---|---|
| | Small | | | Large | | |
| $F_l^r$ | 500 | 1,000 | 2,000 | 500 | 1,000 | 2,000 |
| 20 periods - CPLEX 1H gap | 2.16 | 2.54 | 3.23 | 4.38 | 4.79 | 3.44 |
| 20 periods - SSCG(SSND-RAM) 1H gap | 3.94 | 4.12 | 5.08 | 5.13 | 4.61 | 5.59 |
| | | | | | | |
| 25 periods - CPLEX 1H gap | 4.95(2) | 6.58(1) | 7.69(2) | 7.90(3) | 12.29(2) | 12.15(3) |
| 25 periods - CPLEX 10H gap | 3.00(1) | 3.69 | 4.57 | 4.35 | 4.58 | 5.35 |
| 25 periods - SSCG(SSND-RAM) 1H gap | 4.15 | 5.04 | 6.25 | 5.15 | 5.25 | 6.30 |

Table 8: CPLEX and SSCG regarding to small instances.

What we see in the first two rows is that CPLEX outperforms SSCG(SSND-RAM) on the 20-period instances and that the performance of both methods degrades as the number of owned resources increases. However, on average, the drop-off in performance as the number of resources increases is smaller for SSCG(SSND-RAM) than CPLEX. Turning our attention to the 25-period instances, we see that SSCG(SSND-RAM) significantly outperforms CPLEX in terms of what each can produce in one hour of execution and SSCG(SSND-RAM) is competitive with what CPLEX can produce in ten hours.

We next turn our attention to the performance of SSCG(SSND-RAM) on the larger, rail-based instances (Set 1). These instances are too large for the full set of cycles, $\theta$, to be enumerated in a reasonable running time. As such, to benchmark the ability of SSCG(SSND-RAM) to produce high-quality solutions, we developed an alternative column generation-based heuristic, which we call CGMIP-H. CGMIP-H uses the column generation portion of SSCG(SSND-RAM) to solve the linear programming relaxation of an instance, keeping the cycles generated, $\bar{\theta}$. The heuristic then solves the resulting MIP, SSND-RAM($\bar{\theta}$), with a commercial MIP solver (CPLEX 12.4).

These experiments also span six scenarios defined by three possible values for $F_l^r$ (1,000; 3,000; 5,000) and two fleet sizes (*small*, *large*) derived in a manner similar to what was described above. We execute SSCG(SSND-RAM) for 5 hours or 100 iterations, whichever comes first. When executing CGMIP-H we allow CPLEX 5 hours to solve SSND-RAM($\bar{\theta}$), and do not limit the amount of time spent in the column generation phase. We report the results of these experiments in Table 9. Again, a number in parentheses indicates that CGMIP-H was unable to produce a primal solution. In these results we see that SSCG(SSND-RAM) is superior to CGMIP-H, sometimes producing solutions that are 10% better than CGMIP-H can. As with the small instances reported on in earlier tables, we again see that the performance of both methods degrades as the number of available resources increases.

| | Size of resource fleet | | | | | |
|---|---|---|---|---|---|---|
| | Small | | | Large | | |
| $F_l^r$ | 1,000 | 3,000 | 5,000 | 1,000 | 3,000 | 5,000 |
| SSCG optimality gap | 5.25% | 6.65% | 5.40% | 6.31% | 7.20% | 5.67% |
| CGMIP-H gap | 15.70%(1) | 15.25%(2) | 10.31%(1) | 16.46%(2) | 14.64%(2) | 8.73%(1) |

Table 9: CPLEX and SSCG for medium- and large-size instances.

From these two sets of experiments we conclude that SSCG(SSND-RAM) is capable of producing high-quality solutions and is an effective solution procedure for the SSND-RAM.

# 7 Conclusions and Future Work

We considered a heretofore unstudied planning problem for consolidation carriers; one that links strategic, resource-acquisition and allocation decisions with tactical, service network design-related decisions. In particular, we extended an existing service network design model to support multiple types of resources, and, to not only recognize the need to manage facility-based resources when routing shipments but also acquire new resources and allocate resources to terminals.

We focused on a setting wherein the only rule that must be observed when managing a resource is that it must ultimately return to the terminal to which it is assigned. However, as the rules governing the management of resource schedules are very dependent on the transportation setting, we proposed a model that encodes these rules in appropriately-constructed cycles and thus is adaptable. For many instances of even modest sizes, enumerating the set of cycles that observe the rules governing what a resource may do is too time-consuming. Thus, we presented a solution approach that both generates cycles that appear in high-quality solutions and creates said solutions. The solution approach combines multiple algorithmic techniques, including column generation, slope scaling, intensification and diversification procedures, and exact optimization.

As the problem we studied is, in a sense, "new," we first validated its use with an extensive computational study. We compared the proposed model with other planning approaches for resource acquisition and allocation. Then, having established that our model is worth solving, we turned our attention to whether the solution approach proposed was computationally effective. With another extensive computational study, we found that it was as it outperformed both a commercial mixed integer programming solver and another heuristic for the problem.

The model we proposed is based on the premise that a carrier should balance resource acquisition and allocation costs against the costs incurred transporting customer shipments given those resource-related decisions during a single, representative period of time. Given that these strategic decisions are typically in effect for multiple years, there are likely many periods of customer shipment demand that should be considered when making resource acquisition and allocation decisions. As such, the primary focus of our next work is to incorporate uncertainty into the model; potentially through the use of scenarios. And of course, like the model proposed in this paper, such a model will likely require a customized solution approach.

# Acknowledgments

While working on this project, the fourth author was doctoral student with Département d'informatique et de recherche opérationnelle, Université de Montréal, and student member of CIRRELT. Partial funding for this project has been provided by the Natural Sciences and Engineering Council of Canada (NSERC), through its Discovery Grant program and by the EMME/2-STAN Royalty Research Funds (Dr. Heinz Spiess). We also gratefully acknowledge the support of Fonds de recherche du Québec through their infrastructure grants and of Calcul Québec and Compute Canada through access to their high-performance computing infrastructure.

# References

R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. Network flows: Theory, algorithms, and applications. *Journal of the Operational Research Society*, 45(11):1340–1340, 1994.

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, 2009.

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

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

C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.F. Savelsbergh, and P.H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

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.

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

T.G. Crainic, B. Gendron, and G. Hernu. A slope scaling/lagrangean perturbation heuristic with long-term memory for multicommodity capacitated fixed-charge network design. *Journal of Heuristics*, 10(5):525–545, 2004.

T.G. Crainic, M. Hewitt, M. Toulouse, and D. Vu. Service network design with resource constraints. *Transportation Science //dx.doi.org/10.1287/trsc.2014.0525*, 2014.

R. De Franceschi, M. Fischetti, and P. Toth. A new ILP-based refinement heuristic for vehicle routing problems. *Mathematical Programming B*, 105:471–499, 2006.

G. Desaulniers, J. Desrosiers, and M.M. Solomon, editors. *Column generation*. Springer-Verlag, New York, 2005.

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

F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, Norwell, MA, 1997.

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

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

D. Kim and P.M. Pardalos. A solution to the fixed charge network flow problem using a dynamic slope scaling procedure. *Operations Research Letters*, 24(4):195–203, 1999.

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

J.S. Lewinski. Turbulence ahead: The coming pilot shortage and how it came to be. The Atlantic, July 2014. URL http://www.theatlantic.com/business/archive/2014/07/turbulence-ahead-the-looming-pilot-shortage-and-its-decades-long-history/374171/.

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, 2001a. URL http://www.sciencedirect.com/science/article/pii/S0965856400000057.

S. Melkote and M.S. Daskin. Capacitated facility location/network design problems. *European Journal of Operational Research*, 129(3):481 – 495, 2001b. URL http://www.sciencedirect.com/science/article/pii/S0377221799004646.

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.

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

B. Tita. Railcar shortage in U.S. pushes up lease rates. The Wall STreet Journal, May 2014. URL http://www.wsj.com/articles/railcar-shortage-in-u-s-pushes-up-lease-rates-1401393331.

C. Trudell. Too many cars and not enough trains prompts buffett to expand rail unit. Bloomberg, March 2015. URL http://www.bloomberg.com/news/articles/2015-03-03/buffett-rail-car-shortage-has-toyota-parking-suvs.

D. Vu, T.G. Crainic, and M. Toulouse. A three-phase matheuristic for the capacitated multi-commodity fixed-cost network design with design-balance constraints. *Journal of Heuristics*, 19(5):757–795, 2013.

# 8 Appendix

For completeness, in this section we present the details of the algorithm described in Section 5. As discussed, the solution approach consists of two main components: (1) a column generation procedure for generating new cycles, and, (2) a slope scaling approach for producing primal solutions from a given set of cycles. We next describe those two components in greater detail. At the end of this section we discuss how we calibrated the algorithm.

## 8.1 Column generation

We define SSND-RAM($\bar{\theta}$) to be the SSND-RAM restricted to the cycles in $\bar{\theta}$ and its linear relaxation to be the SSND-RAM with the variables $z^r_\tau$ and $y^{re}_s$ allowed to take on fractional values. We note that due to the presence of the outsourcing variables $y^{re}_s$, SSND-RAM($\bar{\theta}$) is always feasible. As such, the linear relaxation of SSND-RAM($\bar{\theta}$) will be repeatedly solved, with new cycle variables $z^r_\tau$ added to $\bar{\theta}$ when reduced cost calculations indicate that they may lead to an improved LP solution. As in traditional column generation, these reduced cost calculations are based on dual variables associated with constraints that are binding at the current linear programming solution.

Specifically, we associate the dual variables $\gamma^r_l(\leq 0)$ with each constraint in set (3), $\alpha_s(\leq 0)$ with each constraint in set (5), and $\beta_s(\leq 0)$ with each constraint in set (6). The reduced cost of variable $z^r_\tau, \tau \in \theta^r_l$, is then given by the following expression:

$$f^r_l - \gamma^r_l - \sum_{s \in \tau \cap \mathcal{S}} (f^r_s + \alpha_s u^r_s - \beta_s) \tag{11}$$

Thus, given sets of dual variable values, we form an acyclic graph wherein arcs are given the cost $f^r_s + \alpha_s u^r_s - \beta_s$ for each resource $r$. We search for least-cost cycles in this graph with a dynamic programming approach similar to the one presented in Crainic et al. (2014). When the least-cost cycle has a negative reduced cost, we add it to the set $\bar{\theta}$ and solve the SSND-RAM($\bar{\theta}$) again. The procedure repeats until it reaches an iteration wherein no cycles with negative reduced cost are found. At this point, the linear relaxation of the SSND-RAM has been solved. As such, this procedure also yields a bound on the optimal value of the SSND-RAM. However, we also use the cycles generated during this solution process and the solution to the linear relaxation itself to inform a slope scaling procedure that produces high-quality primal solutions. We next describe this slope-scaling procedure.

## 8.2 Slope scaling

When using slope scaling (Kim and Pardalos, 1999; Crainic et al., 2004) to produce solutions to an integer program (such as the SSND-RAM), the approximation problem is typically the linear relaxation of the original problem. Our approximation problem, which we call AP(SSND-RAM), is formed by both relaxing integrality constraints and removing other constraints, leaving what is essentially a multi-commodity flow problem. We note that our slope scaling procedure begins with a fixed set of cycles $\bar{\theta}$ which we construct with a subset of the cycles found during the previously described column generation procedure. Specifically, we populate $\bar{\theta}$ with the cycles $\tau$ such that $z^*_\tau > \alpha$, wherein $z^*_\tau$ indicates the value of the variable in the optimal solution to the linear relaxation of the SSND-RAM and $\alpha$ is an algorithm parameter. The approximation problem we define is similar to what was presented in Crainic et al. (2014) for the SNDRC, which linearized cycle variables $z_\tau$. However, in the SSND-RAM we must also linearize the acquisition and allocation variables, $h^r_{ll'}$, and the outsourcing variables $y^{re}_s$.

To linearize those variables, we reformulate the original formulation by introducing the new 0-1 variable $z^{\tau r}_{ll'}$ which indicates whether cycle $\tau$ is operated by a resource of type $r$ departing from $l'$ with its source from $l$. We have

$$h^r_{ll'} = \sum_{\tau \in \theta^r} z^{r\tau}_{ll'} \tag{12}$$

and

$$z^r_\tau = \sum_{u \in \Lambda} z^{r\tau}_{ll'} \tag{13}$$

for all $\tau \in \theta^r_{l'}$.

Now, AP(SSND-RAM) is defined with two sets of flow variables associated with two sets of decision variables $z^{r\tau}_{ll'}$ and $y^s_{re}$. The first is $x^{k\tau}_{srll''}$, which represents the flow of commodity $k$ on service $s$ that is supported by resource cycle $\tau$ operated by resource type $r$ with the resource sourced from $u$. The second is $x^{rke}_s$, which represents the flow of commodity $k$ on service $s$ that is supported by an outsourced resource type $r$. We note that we only define these new flow variables for arcs that are services as holding arcs, which do not require a resource. We relate these two flow variables to the flow variables of the SSND-RAM, $x^k_s$ for all $k \in \mathcal{K}, s \in S$ with the equation

$$x^k_s = \sum_{r \in \mathcal{R}} \sum_{l,l' \in \Lambda^+} \sum_{\tau \in \theta^r} x^{k\tau}_{srll'} + \sum_{r \in \mathcal{R}} x^{rke}_s, \forall s \in \mathcal{S}, k \in \mathcal{K} \tag{14}$$

Recall that after solving the approximation problem, the values of the variables $(x^{k\tau}_{srll'}, x^{rke}_s)$ are used to create a solution to the SSND-RAM. To approximate the cost of this resulting solution in the AP(SSND-RAM), we define linearization factors for each type of flow variable. Specifically, we associate the linearization factor $\rho^{k\tau}_{srll'}$ with variable $x^{k\tau}_{srll'}$ and $\phi^{rk}_s$ with variable $x^{rke}_s$. The purpose of these linearization factors is for the expression $\sum_{s \in \mathcal{S}} \sum_{k \in \mathcal{K}} \rho^{k\tau}_{srll'} x^{k\tau}_{srll'}$ to approximate the cost $(F^r_{l'} + \sum_{s \in \tau} f^r_s + H^r_{ll'}) z^{r\tau}_{ll'}$ of the resulting solution to the SSND-RAM. Similarly, the expression $\sum_{k \in \mathcal{K}} \phi^{rk}_s x^{rke}_s$ should approximate the cost $f^{re}_s y^{re}_s$ associated with the resulting solution. Formally, the AP(SSND-RAM) is the following optimization problem:

$$\min \left( \sum_{k \in \mathcal{K}} \sum_{a \in \mathcal{A}} x^k_a c^k_a + \sum_{l,l' \in \Lambda^+} \sum_{r \in \mathcal{R}} \sum_{\tau \in \theta^r} \sum_{s \in \mathcal{S}} \sum_{k \in \mathcal{K}} x^{k\tau}_{srll'} \rho^{k\tau}_{srll'}(t) + \sum_{r \in \mathcal{R}} \sum_{s \in \mathcal{S}} \sum_{k \in \mathcal{K}} x^{rke}_s \phi^{kr}_s(t) \right)$$

subject to

$$\sum_{r\in\mathcal{R}}\sum_{l,l'\in\Lambda^+}\sum_{\tau\in\theta^r} x^{k\tau}_{srll'} + \sum_{r\in\mathcal{R}} x^{rke}_s = x^k_s, \forall s \in S,\ k \in \mathcal{K}, \quad (15)$$

$$\sum_{a\in\mathcal{N}^+(i)} x^k_a - \sum_{a\in\mathcal{N}^-(i)} x^k_a = d^k_i, \forall i \in \mathcal{N},\ k \in \mathcal{K}, \quad (16)$$

$$\sum_{k\in\mathcal{K}} x^k_s \leq \max_{r\in\mathcal{R}}\{u^r_s\}, \forall s \in \mathcal{S}, \quad (17)$$

$$x^k_a, x^{k\tau}_{srll'}, x^{rke}_s \geq 0, \forall k \in \mathcal{K}, a \in \mathcal{A}, s \in S, r \in \mathcal{R}, \tau \in \theta^r, l, l' \in \Lambda^+ \quad (18)$$

Solving AP(SSND-RAM) returns two sets of flow values, $\tilde{x}^{k\tau}_{srll'}$, and $\tilde{x}^{rke}_s$. We use these values to create a solution (possibly infeasible) to the SSND-RAM with the following rules:

$$\tilde{z}^{r\tau}_{ll'} = \begin{cases} 1 & \text{if } \sum_{s\in\tau\cap\mathcal{S}}\sum_{k\in\mathcal{K}} \tilde{x}^{k\tau}_{srll'} > 0 \\ 0 & \text{otherwise} \end{cases}, \forall \tau \in \theta^r_{l'}, l \in \Lambda^+, l' \in \Lambda, \quad (19)$$

and

$$\tilde{y}^{re}_s = \begin{cases} 1 & \text{if } \sum_{k\in\mathcal{K}} \tilde{x}^{rke}_s > 0 \\ 0 & \text{otherwise} \end{cases}, \forall s \in \mathcal{S}. \quad (20)$$

The first rule (19) states that resource cycle $\tau$, which requires a resource type $r$ assigned to terminal $l'$ and sourced from terminal $l$ will be used if there is commodity flow on any of the services it supports. The second rule (20) states that a service will be operated by a third party resource if it transports any commodity flow. However, we note that such solutions may not be feasible to the SSND-RAM as they may violate constraints from the sets (5, 6, 3 and 2). We next discuss how we update the linearization factors, $\rho^{k\tau}_{srll'}$ and $\phi^{rk}_s$ to reflect the fixed cost of the solution.

Solving the AP(LWNSD) yields vector $\tilde{x}^{k\tau}_{srll'}(t)$ which can be used to derive a (possibly infeasible) solution to SSND-RAM by setting $\tilde{z}^{\tau r}_{ll'}(t) = 1$ if $\sum_{k\in\mathcal{K}}\sum_{s\in\tau} \tilde{x}^{k\tau}_{srll'} > 0$ and $\tilde{y}^{re}_s(t) = 1$ if $\sum_{k\in\mathcal{K}} \tilde{x}^{rke}_s > 0$.

Using equations (12), (13), (14) and the objective function (1), for each cycle $\tau \in \theta^r$, the values of $\rho^{k\tau}_{srll'}(t+1)$ are calculated to satisfy the relation:

$$\sum_{k\in\mathcal{K}}\sum_{s\in\mathcal{S}} \tilde{x}^{k\tau}_{srll'}(t)\rho^{k\tau}_{sr}(t+1) = \left(F^r_l + \sum_{s\in S\cap\tau} f^r_s + H^r_{ll'}\right)\tilde{z}^{r\tau}_{ll'}(t), \forall r \in \mathcal{R}, \tau \in \theta^r, \quad (21)$$

which can be done by setting

$$\rho^{k\tau}_{srll'}(t+1) = \begin{cases} \frac{F^r_l + \sum_{s\in\tau\cap S} f^r_s + H^r_{ll'}}{\sum_{k\in\mathcal{K}}\sum_{s\in\tau\cap S} \tilde{x}^{k\tau}_{srll'}} & \text{if } \sum_{k\in\mathcal{K}}\sum_{s\in\tau\cap S} \tilde{x}^{k\tau}_{srll'} > 0, \\ \rho^{k\tau}_{srll'}(t) & \text{otherwise,} \end{cases} \quad (22)$$

and for each outsourced service arc $y^{re}_s$, the values of $\phi^{rk}_s$(t) satisfy:

$$\sum_{k\in\mathcal{K}} \tilde{x}^{rke}_s \phi^{rk}_s(t) = \tilde{y}^{re}_s(t) f^{re}_s, \quad (23)$$

which can be done by setting:

$$\phi_s^{rk}(t) = \begin{cases} \frac{f_s^{re}}{\sum_{k\in\mathcal{K}} \tilde{x}_s^{rke}} & \text{if } \sum_{k\in\mathcal{K}} \tilde{x}_s^{rke} > 0, \\ \phi_s^{rk}(t-1) & \text{otherwise.} \end{cases}$$

When the solution $(\tilde{x}, \tilde{z}, \tilde{y})$ to AP(SSND-RAM) violates a constraint from one of sets (5, 6, 3, and 2), our approach next executes the following procedures to try and construct a feasible solution to SSND-RAM.

## 8.3 Creating a feasible solution to SSND-RAM from a solution to AP(SSND-RAM)

Because the AP(SSND-RAM) does not include all the constraints that are present in the SSND-RAM (constraint sets 2, 3, 5, and 6), and unlike most slope scaling approaches, rounding procedures are not sufficient to construct a feasible solution to the SSND-RAM from a solution, $(\tilde{x}, \tilde{z}, \tilde{y})$, to the AP(SSND-RAM). Instead, we must execute another procedure to construct a feasible solution to the SNDRC. Our procedure, instead of modifying the solution to the AP(SSND-RAM) directly, creates a subgraph, $\bar{\mathcal{G}}^r$ of $\mathcal{G}$ that contains the nodes $\mathcal{N}$ and a subset, $\bar{\mathcal{A}}^r$, of the arcs $\mathcal{A}$ that can be decomposed into cycles. Then, the procedure extracts cycles from this subgraph that can be used to construct a feasible solution to the SSND-RAM.

For each $r$, we create $\bar{\mathcal{G}}^r$ by first adding to $\bar{\mathcal{A}}^r$ all service arcs $s \in \mathcal{S}$ such that either $\sum_{k\in\mathcal{K}} \tilde{x}_{srll'}^{k\tau} > 0$ for some $k \in \mathcal{K}, l \in \Lambda^+, l' \in \Lambda$, or they belong to a cycle used in the AP(SSND-RAM) solution (i.e., $\tilde{z}_{ll'}^{r\tau} = 1$). We also add all holding arcs to $\bar{\mathcal{A}}^r$. We then solve an optimization problem to add service arcs in $\mathcal{A} \setminus \bar{\mathcal{A}}^r$ to $\bar{\mathcal{A}}^r$ to ensure that $\bar{\mathcal{G}}^r$ can be decomposed into cycles in such a way that each service arc appears in at most one cycle but holding arcs may appear in multiple cycles, as doing so maximizes the number of cycles that can be extracted from $\bar{\mathcal{G}}^r$. The objective of this optimization problem is to minimize the total cost of the arcs added with respect to the cost coefficient $f_s^r$. We note that we formulate and solve this optimization problem as a minimum cost, maximum flow problem (Ahuja et al., 1994). See Vu et al. (2013) for details of how a similar procedure was done for a network design problem with Eulerian-type constraints.

After creating $\bar{\mathcal{G}}^r$, we next extract a set of cycles $\tilde{\theta}^r$ from this network that are guaranteed to satisfy Constraints (6), which ensure that each service is executed at most once. Then, we formulate and solve SSND-RAM($\tilde{\theta}$), which is the SSND-RAM restricted to the cycles in the set $\tilde{\theta} = \cup_{r\in\mathcal{R}}\tilde{\theta}^r$. We set a time limit on the solution of this mixed integer program of $t_{MIP}$ seconds. We note that this procedure is guaranteed to find a feasible solution to the SSND-RAM in each iteration.

## 8.4 Intensification and Diversification

Metaheuristics (Glover and Laguna, 1997) often include intensification procedures, wherein a promising region of the solution space is explored deeply, and diversification procedures, wherein the search is directed towards regions of the solution space that have not yet been thoroughly searched. As these procedures often enable the search to find high-quality solutions in limited times, we have included them in our solution approach. We first present our intensification procedure and then the diversification procedure.

For intensification, and similar to how we create a feasible solution to SSND-RAM from a solution to AP(SSND-RAM), we introduce an exact optimization step into the search, wherein SSND-RAM($\tilde{\theta}$) is solved with a commercial mixed integer programming solver. Like above, we again set a time limit on the solution of this MIP of $t_{MIP}$. Our intensification procedure differs from how we create a solution to SSND-RAM from one to AP(SSND-RAM) in how the set of cycles $\tilde{\theta}$ is created. Whereas previously the set $\tilde{\theta}$ was derived from the services that appeared in the most recent solution to the AP(SSND-RAM), our intensification procedure derives $\tilde{\theta}$ from the services that appeared in the last $q$ solutions of the AP(SSND-RAM). The rest of the procedure is as described above. The intensification procedure is executed when an improved solution has not been found after a predefined number of iterations.

For diversification, because it is the cycles and outsourced services in the solution to AP(SSND-RAM) at a given iteration that dictate the structure of the resulting solution to SSND-RAM, we periodically modify the objective function of AP(SSND-RAM) to avoid frequently used cycles and outsourced services. To do so, we collect the number of times, $fre_\tau$, each cycle $\tau$ appears in a solution to AP(SSND-RAM). Similarly, we also collect the number of times, $fre_s$, each outsourced service arc $s \in \mathcal{S}$ appears in a solution to AP(SSND-RAM). Then, if the diversification condition is met for each cycle $\tau \in \bar{\theta}^r$ and outsourced service arc $s$ in the current solution of the approximated problem AP(SSND-RAM), we set the linearization factor $\rho^{k\tau}_{srll'}(t)$ to $\rho^{k\tau}_{srll'}(t)(1 + \epsilon * fre_\tau)$ and $\phi^{rk}_s(t) = \phi^{rk}_s(t)(1 + \epsilon * fre_s)$, where $\epsilon$ is an algorithm parameter.

We continue to update $\rho(t)$ and $\phi(t)$ in this manner for a fixed number of iterations that is dictated by the algorithm parameter $I^{diver}_{max}$. Lastly, it is possible for two consecutive solutions of AP(SSND-RAM) to be the same, in which case the slope-scaling procedure will terminate. To continue the execution of slope-scaling when this occurs, we add a penalty value $P$ to the linearization factors as seen in (24) and (25). In our experiments, $P$ is set to the objective value of the current solution to AP(SSND-RAM).

$$\rho^{k\tau}_{srll'}(t+1) = \begin{cases} \frac{P + H^r_{ll'} + F^r_l + \sum_{s\in\tau\cap\mathcal{S}} f^r_s}{\sum_{k\in\mathcal{K}}\sum_{s\in\tau\cap\mathcal{S}} \tilde{x}^{k\tau}_{srll'}} & \text{if } \sum_{s\in\tau\cap\mathcal{S}}\sum_{k\in\mathcal{K}} \tilde{x}^{k\tau}_{srll'} > 0 \\ \rho^{k\tau}_{srll'}(t) & \text{otherwise} \end{cases} \tag{24}$$

$$\phi^{rk}_s(t+1) = \begin{cases} \frac{P + f^{re}_s}{\sum_{k\in\mathcal{K}} \tilde{x}^{rke}_s} & \text{if } \sum_{k\in\mathcal{K}} \tilde{x}^{rke}_s > 0 \\ \phi^{rk}_s(t) & \text{otherwise} \end{cases} \tag{25}$$

## 8.5 Calibrating SSCG(SSND-RAM)

The solution approach proposed, SSCG(SSND-RAM), is governed by four parameters. We list in Table 10 these parameters, as well as the values that have been considered when calibrating the algorithm. We calibrated the algorithm by executing it for each of the 16 possible parameter value combinations on instance Set 2 (described in Table 2). To make the selection, we computed averages of the optimality gap and execution time for each combination over those 14 instances. We found that the difference between the best and worst among the ten best average gaps is about .7%, suggesting that the algorithm is quite robust with respect to the values of these parameters. We list the value chosen for each parameter in Table 10.

| Parameter | Description | Values tested | Value selected |
|---|---|---|---|
| $I_{max}^{diver}$ | Number of diversification iterations | 3, 5 | 3 |
| $\epsilon$ | The effect of frequency on linearization factor | 0.5, 1 | 1 |
| $t_{MIP}$ | Time limit $t_{MIP}$ for solving a SSND-RAM by the solver | 600, 900 | 600 |
| $\alpha$ | Parameter for cycle's selection | 0.01, 0.1 | 0.1 |

Table 10: Algorithm parameters and calibration settings

Lastly, we calculate "Utilization", as reported in Tables 4 and 6 as

$$\left(\sum_{s\in S_{OBO}}\sum_{k\in\mathcal{K}} x_s^k\right) / \left(\sum_{s\in S_{OBO}}\sum_{r\in\mathcal{R}} u_s^r\right), \tag{26}$$

where $S_{OBO}$ is the set of executed services that are supported by owned resources.