Motor Carrier Service Network Design

Bakir, Ilke; Erera, Alan; Savelsbergh, Martin

Published in:
Network Design with Applications to Transportation and Logistics

DOI:
10.1007/978-3-030-64018-7_14

Publication date:
2021

Chapter 14
Motor Carrier Service Network Design

Ilke Bakir, Alan Erera, and Martin Savelsbergh

1 Introduction

The trucking, or motor freight, industry provides ground freight transportation services to shippers using road trucks. Motor carriers provide multiple types of services, differentiated to serve shipments with different characteristics. *Truckload* services are offered to shippers who move dedicated trailers or containers each directly from an origin location to a destination location. Truckload services are provided both by large firms with thousands of tractors and trailers but also by small companies that may sometimes operate fleets with only a few vehicles. In contrast, *consolidation trucking carriers* operate both a network of freight transfer terminals and also a fleet of vehicles to provide a schedule of transportation services for shippers moving smaller quantities. There are two primary consolidation service types. *Less-than-truckload* (LTL), or freight, services provide shippers with the capability to send smaller shipments that do not require an entire trailer; an LTL carrier consolidates shipments into truckload movements between terminals to provide cost-effective service. *Package* services serve shippers seeking to move the smallest shipments, typically letters, small parcels, and boxes.

Truck transportation in most countries is currently the dominant land transportation mode, accounting for the largest fraction of revenue and moving the most tons. For example, in the United States in 2016 trucking accounted for 63% of

I. Bakir
Department of Operations, Faculty of Economics and Business, University of Groningen, Groningen, Netherlands
e-mail: i.bakir@rug.nl

A. Erera (✉) · M. Savelsbergh
H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA
e-mail: alan.erera@isye.gatech.edu; martin.savelsbergh@isye.gatech.edu

T. G. Crainic et al. (eds.), *Network Design with Applications to Transportation and Logistics*, https://doi.org/10.1007/978-3-030-64018-7_14

the total tonnage moved and 62% of the total value of all shipments (Bureau of Transportation Statistics 2018); in the European Union and in Asia, motor freight is similarly important. Trucking services provide fast *transit times* to shippers with only air freight able to provide shorter times. Transit time is a primary measure of *customer service level* in shipping, and many modern freight services guarantee transit times to shippers.

This chapter focuses on service network design for consolidation trucking carriers. Optimization models and solution approaches for the core network design problems, which include flow and load planning, will be covered in detail. Other related operations design problems will be discussed more briefly. The goal is to provide a thorough introduction to these problems and methods and to focus the discussion on ideas that have had an impact on practice. The chapter will also highlight some of the newest work in this area and help guide researchers beginning work in this field.

The remainder of the chapter is organized as follows. Section 2 provides an overview of trucking operations, focusing specifically on the structure of the networks operated by consolidation trucking carriers. Section 3 introduces models for trucking network flow planning and describes exact and heuristic approaches for building solutions to these models. Section 4 then describes integrated flow and load planning models that rely on time-expanded networks and describes large-scale local search heuristics for their solution. Section 5 briefly describes key developments in the trucking service network design literature. Finally, Sect. 6 provides some perspective on the current state of this research area and discusses a number of ongoing research trends that hold promise for this field.

2 Consolidation Trucking Operations

Consolidation trucking carriers plan and operate service networks to provide freight transportation services directly to shippers seeking to move less-than-truckload or package freight. Carriers establish a geographic region within which they will operate, and more specifically determine origin and destination pairs between which they will provide service and for which categories (or *classes*) of freight. Each service offering for an origin-destination pair also includes a price (or freight *rate*) and a transit time. In some cases, transit times provide only a rough estimate of the number of days required for the execution of the transportation service, while in other cases time-definite offerings specify precisely how long a shipment will require and when it will arrive (for example, 2-day or next-morning).

Given a set of service offerings, a consolidation trucking carrier must build and operate a service network to satisfy customer demand feasibly and cost-effectively. To do so, a medium to large carrier operates a network of transfer *terminals*. Trucking terminals have facilities for truck loading and unloading; these *docks* enable rear-loading trucks to park with the trailer deck at the same level of the terminal floor. In LTL operations, the truck trailers that are used to make pickups and deliveries at customer locations are similar or identical to those used for the long distance terminal-to-terminal movements. In package operations, smaller delivery vans are used when visiting customers, and terminals therefore may have different loading areas for different truck types.

All terminals have the capability to *sort* freight shipments to be loaded into different outbound truck trailers or containers. *Cross-dock sorting*, or *cross-docking*, is a sorting system where larger shipments (often on pallets or within intermediate containers) are moved from unloading trailers to loading trailers by forklift or pallet jack. The name cross-docking refers to the fact that shipments are moved directly "across the dock" and are not stored in any intermediate locations. Cross-docking is the primary sorting operation used by LTL carriers. Since parcel shipments are smaller, terminals operated by package carriers typically include automated and/or manual piece-sorting equipment. Examples of piece-sorting equipment include cross-belt sorters or manual sorting cabinets. Smaller packages, parcels, and letters may be consolidated into bags or other types of intermediate containers before they are loaded into trailers. Package carriers may also use conveyor belt systems and additional belt sorters to enable movement of parcels through the terminal as well as to facilitate cross-docking of larger parcels, bags, and intermediate containers.

Thus, the primary role of terminals in trucking networks is consolidation of smaller shipments into truckloads and the related transfer of shipments between inbound and outbound trailers and containers. Consolidation and transfer allows a trucking carrier to provide cost-effective service between large numbers of origins and destinations. For example, a carrier that operates n terminals with direct service between all pairs would need to move trailers on $n(n-1)$ service lanes, but if one of the central locations were used as a transfer hub this number could be reduced to as few as $2(n-1)$ lanes. Individual truckload dispatches in a well-designed consolidation network will have higher trailer utilization and the total required trailer-miles required to move freight from origins to destinations should decrease. However, each individual shipment may travel farther (thus increasing system ton-miles) and may be sorted one or more times at intermediate transfer terminals.

A typical consolidation terminal network is depicted in Fig. 14.1, in this case for an LTL carrier. Carriers typically operate two types of terminals. An *end-of-line*, or satellite, terminal is a smaller facility that only enables transfer of freight between the *pickup-and-delivery* operation and the *linehaul* operation. A *hub*, or breakbulk, terminal is a larger facility that provides both the functionality of an end-of-line terminal while also providing transfer opportunities between terminals in the linehaul network. Each end-of-line terminal may be connected with dispatches to and from only a small set of hub terminals, while hub terminals may provide dispatches to and from a large number of end-of-line terminals and other hubs. Package networks have similar designs.

Effective freight transfer also requires timed coordination of unloading, sorting, and loading activities at transfer terminals. For this reason many carriers divide each operating day into distinct *sorting periods* or, more simply, *sorts*. Trailers arriving for a sort are unloaded and the freight shipments are sorted into outbound trailers for dispatch by the conclusion of the sort. It is quite common for terminals to operate a

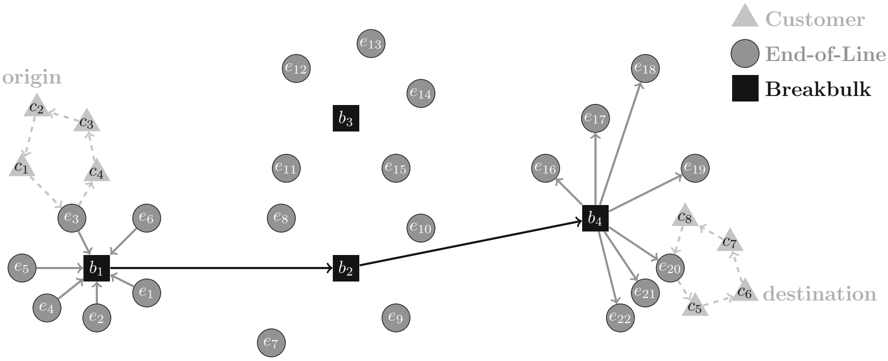

Fig. 14.1 A network of an LTL trucking carrier, serving customer locations using end-of-line and breakbulk cross-dock terminals

morning sort and an evening sort, while larger transfer terminals may also operate additional overnight and midday sorting periods.

Given a network of terminals, a consolidation trucking carrier will operate a pickup-and-delivery operation and linehaul operation. The pickup-and-delivery operation is used to collect freight from customers and transport it to an origin consolidation terminal and to distribute freight for the last-mile from the final terminal to destinations. In package systems, the vehicles used for pickup and delivery are usually smaller delivery vans while LTL carriers often use short or medium length trailers. Pickup and delivery operations require the execution of multi-stop routes with significant time constraints; customer facilities have time windows when they can send or receive shipments, and the carrier has deadlines when freight must leave from or arrive to a terminal in order to meet the service expectation of the customer. It should be noted that high-volume shippers often interface with consolidation carriers by using *drop shipping* or by supplying dedicated customer trailer equipment. In these scenarios, the shipper or the carrier may move truckloads of shipments directly into the linehaul network at a carrier terminal thus skipping the traditional pickup process.

A carrier linehaul network operation provides transportation of truck trailers and containers between its transfer terminals. A *load* in an LTL or package network refers to a trailer or a container that is loaded and dispatched from an origin terminal and headed for unloading at a destination terminal. For LTL carriers, loads are most frequently moved by company truck drivers using either long trailers or trains of two or three short trailers. Short trailers, such as 28-foot pups in the US, provide the carrier with the ability to have single drivers move multiple loads with smaller individual volumes simultaneously. Linehaul movements of loads between terminals, or *dispatches*, tend to be short to allow drivers to return to their home terminals within a single operating day. When loads are created between distant terminals, they frequently are not moved directly by a single driver. Instead, the load may be transferred using two or more *movements*, where each movement is executed either by a company driver or by an outside contractor. The intermediate stops in these sequences of movements are typically called *relays*, and they may occur at terminals or at dedicated relay facilities; operating in this way can both speed the movement of long distance loads while also eliminating the need for some intermediate sorting. For certain long distance loads, freight railroads may be used to move trailers or rail-compatible containers in an outsourcing arrangement; such rail *intermodal* movements are less costly but require longer travel times and may introduce more travel time uncertainty.

2.1 Trucking Service Network Design Problems

In consolidation trucking systems, we refer to the service network as the set of transportation and supporting activities operated by a trucking carrier in order to provide transportation services to shippers. If we think of a network using its general definition as a set of interacting components, then a service network operated by a trucking firm refers to truck transportation movements and associated loading, sorting, and unloading activities. *Service network design* problems in truck transportation focus on building designs and operational plans for these networks; see Crainic (2000) for a comprehensive review of earlier work in all areas of freight transportation. Typically, *physical network* design questions such as determining the type, number, and size of terminal facilities to operate are not considered service network design problems. There is a significant body of literature in facility location (see e.g., Love et al. 1988; Mirchandani and Francis 1990; Drezner and Hamacher 2001; Snyder 2006; Daskin 2011) and a subset that focuses specifically on the location of truck transportation terminals known as hub location problems (see e.g., O'Kelly 1986; Campbell 1994; Alumur and Kara 2008; Farahani et al. 2013), and thus we will ignore these problems in this chapter.

Since trucking service networks can be complex and require many design and planning decisions, a large number of problems could be classified as service network design problems including:

- flow planning problems;
- load planning, routing, and dispatch problems;
- driver and equipment fleet management problems; and
- vehicle routing and scheduling problems.

Flow planning, or freight routing, problems seek to determine how shipments should flow, or be routed, through a terminal network en route from origin to destination. A freight route for an individual shipment specifies the sequence of terminals where the shipment will be transferred via cross-docking or other sorting methods; in most cases, a shipment is unloaded from one truck and reloaded onto another at each of these terminal stops. While it is possible to dynamically determine a freight route for each contracted shipment, it is much more common for carriers to establish a fixed flow plan that specifies a route for each shipment given its origin and destination terminals and its service requirements. This chapter will focus primarily on flow planning problems since they are in some sense the core service network design problem in trucking. Other problems will be considered when they also include some flow planning component, as we describe now.

Load planning, routing, and dispatch problems seek to determine how to build consolidated freight loads from shipments and time their dispatch. For trucking carriers, a load will be a trailerload or a containerload. Load planning problems can be tactical or operational. At the tactical level, a consolidation carrier would like to determine how many loads (of potentially different sizes) need to be dispatched between terminals, and at what times, in order to feasibly serve the demands induced at the flow planning step. When a load is planned between more distant terminals, it is also necessary to determine a movement path for the trailer through the network if it is not to be dispatched directly from its origin to destination terminal. This load routing step determines the sequence of relay points visited by the load and the transportation mode used for each connecting movement leg. When carriers dispatch trains of short trailers, like two pup combinations, it is also necessary to determine which loads to pair up into combinations when routing loads. At the operational level, loads need to be constructed from actual available shipments; often, loads may be cancelled or added on the day of operations, or shipments shifted onto alternate freight routes, to serve demands and utilize transportation capacity most cost-effectively. Modern service network design approaches often integrate flow and load planning rather than treat the problems sequentially; in such models, freight routing decisions and timed load dispatching decisions are made simultaneously. Additionally, operational models for load planning and dispatching may also allow limited flow replanning choices.

It is useful to note here that LTL and package express carriers are not the only firms that need to solve flow and load planning problems. Large shippers and 3PL companies often face flow and load planning problems when designing consolidation operations for distribution networks. Less-than-truckload shipments for such companies can be consolidated and routed through cross-docking facilities or pool points to avoid outsourcing to LTL carriers. Given a network design, such companies use truckload carriers to provide the trucking movements. It is also somewhat common in these cases for shippers to use multi-stop truckload movements referred to as “milk runs”. In this scenario, a shipper loads a truckload trailer at a single origin to be delivered to a sequence of partial load drop-off locations (or alternately might load at multiple pickup locations before moving the trailer to a single final destination for unloading).

Driver and equipment fleet management problems focus on building plans and schedules that enable trucking loads to be executed. Planned loads must be loaded into appropriate equipment, typically trailers or containers that have specific capabilities. Empty equipment repositioning problems are used to ensure that empty trailers and containers of the required equipment types are available over time where needed, and models for flow and load planning are more frequently now including constraints on equipment balance and availability. All trucking loads are moved at some point during their journeys by one or more truck drivers. Truck drivers must be managed to not violate government work regulations and sometimes are also subject to employee union restrictions. Consolidation carriers typically operate driver schedules that also must be planned in advance. Incorporating driver management decisions into flow and load planning models is often a difficult challenge because of the complicated nature of driver constraints.

Finally, vehicle routing and scheduling problems may also be considered service network design problems. For example, the classical capacitated vehicle routing can be described as a one-to-many load planning problem with vehicle resources that must operate on customer-disjoint cycles from a single depot; given this setup, the unique flow decision for each depot-to-customer shipment is to move from the depot along the route serving the customer, visiting intermediate stops as necessary before arriving to the destination. Consolidation carriers operate last-mile pickup and delivery operations that bring shipments from customer origin locations into first-level terminals at the beginning of trips and then distribute them at the end of trips, and thus they face specific vehicle routing and scheduling problems. Since the literature on last-mile truck vehicle routing and scheduling problems is vast, we will not cover it in this chapter. The reader is instead referred to excellent recent survey papers covering the area (see e.g., Golden et al. 2008; Cattaruzza et al. 2017; Braekers et al. 2016; Savelsbergh and van Woensel 2016; Psaraftis et al. 2016).

3 Network Design Models for Flow Planning

We begin with flow planning problems, the core service network design problems faced by LTL and package trucking carriers. The goal of flow planning problems is to determine a plan for consolidation of shipments into flows between transfer terminals to take advantage of certain cost scale economies in transportation. As an introduction, we begin by describing the components of network design mixed-integer programs for flow planning. To do so, we start with a base model of geographic consolidation.

Given a terminal set N, the freight carrier faces the problem of deciding how to transfer freight that originates at some terminal $o \in N$ and is destined for another terminal $d \in N$. We use the term *commodity* to describe such freight, and we let K be the set of all commodities to be moved by the carrier. Suppose that commodity k originates at terminal o_k and is destined for terminal d_k, and let q_k be a measure of the volume (or *flow*) of freight to be transferred. Note that a commodity represents the aggregation of shipments for many customers, and q_k measures this aggregated volume. Furthermore, suppose for now that at most one commodity is defined with the same origin-destination pair (o, d); this is possible, for example, when all shipments moving from o_k to d_k are promised the same transit time. Note here that volume or flow is a rate: a quantity moving (or to be moved) per time.

Typical units of measure for freight flow in trucking are pounds per day or tons per week, but it is important for flow planning models to know how this freight flow converts to the number of truck trailerloads necessary to move the volume.

During operations, detailed information about the size and weight of each shipment is used when determining how to pack and load trailers feasibly and effectively but this information is not known with certainty at the planning stage. For simplicity, it is common instead to convert estimated freight flows into an equivalent number of trailers by using simple factors (for example, with units of trailers per pound). It may be reasonably accurate to use a network-wide conversion factor for this task, however, the mix of freight shipment types (and their associated weight per cubic volume densities) may vary on different origin-destination lanes and thus it may be necessary to use different conversion factors on different lanes.

A *flow plan* is a set of decisions that specifies jointly how all commodities should be transferred from origins to destinations cost-effectively while meeting customer service requirements, the most important of which is the transit time. The simplest flow planning decision for a commodity is to move it in *direct trailers* or containers, loaded at the origin terminal o_k and unloaded at the destination d_k. We refer to this decision as a *direct route* for commodity k. Note that the use of the word "direct" in this context refers to the fact that the freight for this commodity will not be unloaded, sorted, and reloaded at any intermediate hub terminals. However, a direct trailer or container from terminal i to j may certainly be transported by a sequence of movements, by multiple drivers through relay points, or even by using multiple modes of transportation. A trivial and likely expensive flow plan would be to move all commodities along direct routes; note that given enough driver and trailer resources, this direct route flow plan should also be service feasible since there is no faster way to transfer freight between origins and destinations.

Consider then the non-trivial case where some commodities will not be assigned to direct freight routes. Let A be a set of directed arcs where $(i, j) \in A$ models a lane where trailers (or containers) can be loaded at terminal $i \in N$ and moved to terminal $j \in N$ for unloading. In a physical network with hub (N_H) and end-of-line (N_E) terminals, such load arcs (i, j) should exist between all pairs of hub terminals in N_H. On the other hand, when i or j is an end-of-line in N_E, it may be possible to reduce the number of arcs in a network by restricting the generation of direct loads to or from a limited set of terminals. Care should be exercised when doing so, however, since it may be more sensible to allow a model to decide where to build loads. Recall again that a direct load (i, j) does not imply that trailers are moved from i to j with a single driver or by a single movement.

Given A, let p_k be a possible freight route (or *path*) from o_k to d_k. Using the typical definitions from mathematical networks, each p_k is a simple path: a connected sequence of arcs in A beginning at node o_k and ending at node d_k with no cycles. For convenience, p_k may also be used to refer to a sequence of nodes in N where the initial node is o_k, the final node is d_k, and an arc $a \in A$ exists between each pair of adjacent nodes in the sequence. Let P_k be the set of all freight paths in A that connect o_k to d_k. Using these ideas, the primary decisions in every flow planning problem are to assign commodity flow to one or more feasible paths $p_k \in P_k$ for each commodity $k \in K$ to minimize logistics costs while meeting service requirements. Referring again to Fig. 14.1, a path from one of the end-of-line terminals on the left to one on the right, e.g., $c_2 \rightarrow c_1 \rightarrow e_3 \rightarrow b_1 \rightarrow b_2 \rightarrow$

$b_4 \rightarrow e_{20} \rightarrow c_5 \rightarrow c_6$, represents a freight path for that commodity, where a cross-dock transfer occurs at the head node of each arc in the path.

The remainder of this section will develop flow planning optimization formulations using *flat* network models, which we distinguish from *time-space* or *time-expanded* networks which model both geographic locations and explicit decision timing. Flat network models have the advantage that they lead to smaller integer programming instances, but they provide a relatively coarse approximation of trucking operations that is most useful for tactical planning. The input demands q_k represent average flow rates per time (e.g., tons or pallets or equivalent trailers per week) and the output freight and equipment decision variables also will represent flow rates per time. For this reason, it is natural to refer to such models as *rate-based*.

3.1 Arc-Based Flow Planning Model for Consolidation Trucking

To build a flow planning model, we make a few assumptions. Suppose that all shipments using truck movement lane $(i, j) \in A$ are loaded into a trailer or container at i and unloaded and sorted at j, and furthermore that the costs of transportation are separable by lane and the costs of sorting are separable by terminal. Finally, suppose that commodity flow rates q_k are roughly constant over time, and that any timing issues regarding consolidation can be safely ignored. Let x_{ij}^k be a decision variable representing the flow of commodity k moving on lane $(i, j) \in A$, measured in the same units as q_k. A generic mathematical programming formulation for flow planning is then:

$$\text{minimize} \quad \sum_{(i,j)\in A} f_{ij}^T(x_{ij}) + \sum_{i\in N} f_i^H(x_{i*}) \tag{14.1}$$

subject to

$$\sum_{(i,j)\in A} x_{ij}^k - \sum_{(j,i)\in A} x_{ji}^k = \begin{cases} q_k & \text{if } i = o_k \\ -q_k & \text{if } i = d_k \\ 0 & \text{otherwise} \end{cases} \qquad \forall k \in K,\ \forall i \in N \tag{14.2}$$

$$x_{ij} = \sum_{k\in K} x_{ij}^k \qquad \forall (i, j) \in A \tag{14.3}$$

$$x_{i*} = \sum_{k\in K \mid o_k \neq i} \sum_{(i,j)\in A} x_{ij} \qquad \forall i \in N \tag{14.4}$$

$$x_{ij}^k \geq 0 \qquad \forall k \in K,\ \forall (i, j) \in A \tag{14.5}$$

Constraints (14.2) and (14.5) define a simple (uncapacitated) case of the linear multi-commodity flow polytope, and this formulation allows freight flow to be split across many paths for each commodity. The objective function here is specified generically, where transportation costs are separable by lane (i, j) and handling costs are separable by terminal i. We now discuss how typical objective functions lead to network design mixed-integer programs.

First, it is most common in flow planning models for trucking terminal handling costs to be modeled as linear in throughput,

$$f_i^H(x_{i*}) = h_i x_{i*}, \tag{14.6}$$

where h_i is the handling cost rate per flow unit and x_{i*} is the total freight volume handled at terminal i. Handling in flow planning refers to the transfer of freight either via cross-docking of larger shipments or piece sorting in parcel operations. It is reasonable to assume that sorting labor cost or equipment operating cost grows roughly linearly with freight volume in flow planning models. Including handling costs in flow planning models explores a tradeoff with transportation costs; thus, it is quite common to estimate handling cost rates (which can be hard to measure precisely) to strike a reasonable balance with truck transportation costs.

Second, the truck transportation cost function on each arc should exhibit some cost economies scale in flow, *i.e.,* the average cost $\frac{f_{ij}^T(x)}{x}$ should be decreasing for at least some values of x to encourage consolidation. Note that with linear handling costs and linear transportation costs $c_{ij}x_{ij}$, the flow planning problem can be solved simply by finding a minimum cost path for a unit flow from o_k to d_k for each commodity and then moving all flow q_k along this path.

A reasonable approach for estimating truck transportation costs might be to assume a fixed cost d_{ij} for the dispatch of each unit trailerload (or containerload) on lane (i, j). Suppose all trailers have the same *capacity*, *i.e.,* once a trailer contains Q units of flow an additional trailer is needed. Then, f_{ij}^T is the following step function:

$$f_{ij}^T(x) = d_{ij}\left\lceil \frac{x}{Q} \right\rceil, \tag{14.7}$$

where the ceiling function rounds the value of x up to the next unit load. In practice, it is common to simply measure q_k in fractional trailers and to set $Q = 1$. Since the width of each step is always Q units and the height is always d_{ij}, it is straightforward to use an integer variable τ_{ij} to model this step function by simply forcing $\tau_{ij} \geq \frac{x_{ij}}{Q}$. It is important to make note of a few ideas when modeling transportation costs with per trailer lane costs d_{ij}. To estimate d_{ij} accurately requires that we know the movement (relay) path for the load from i to j in advance; it is common to use the most frequently used such path. Furthermore, since LTL and package carriers often dispatch trains of two short trailers together, this approach also is most accurate for carriers where dispatches almost never move short trailers alone; in such scenarios, d_{ij} represents one-half of the cost of moving a two-trailer train from i to j.

It may also be useful to model some transportation cost beyond the fixed cost per trailer that accrues linearly with flow, for example to account for fuel and maintenance costs that may increase with transported load size. In this case, define a linear arc flow cost as

$$f_{ij}^{L}(x) = c_{ij}x = (h_i + c_{ij}^{T})x,$$

and now define the flow planning problem as the following mixed integer linear programming problem:

$$\text{minimize} \quad \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ij}^{k} + \sum_{(i,j) \in A} d_{ij} \tau_{ij} - \sum_{i \in N} \sum_{k \in K \mid o_k = i} h_i q_k \tag{14.8}$$

subject to

$$\sum_{(i,j) \in A} x_{ij}^{k} - \sum_{(j,i) \in A} x_{ji}^{k} = \begin{cases} q_k & \text{if } i = o_k \\ -q_k & \text{if } i = d_k \\ 0 & \text{otherwise} \end{cases} \qquad \forall k \in K, \ \forall i \in N \tag{14.9}$$

$$x_{ij} = \sum_{k \in K} x_{ij}^{k} \qquad \forall (i,j) \in A \tag{14.10}$$

$$x_{ij} \leq Q\tau_{ij} \qquad \forall (i,j) \in A \tag{14.11}$$

$$x_{ij}^{k} \geq 0 \qquad \forall k \in K, \ \forall (i,j) \in A \tag{14.12}$$

$$\tau_{ij} \geq 0 \text{ and integer} \qquad \forall (i,j) \in A \tag{14.13}$$

The non-negative integer variable τ_{ij} measures the minimum required trailers on lane (i, j) when (14.11) is matched with the positive objective function coefficient d_{ij}, thus properly modeling the lane step function dispatch costs given by (14.7). Note also that the objective function subtracts off a constant to avoid paying handling costs at terminals where freight originates; of course, including this constant or any other in the objective function does not affect the flow plan, only its computed cost. In the following subsections, we will no longer include such objective function constants in the flow planning formulations.

This generic arc-based flow planning model is a multi-commodity capacitated fixed-charge network design (MCND) problem. Solving this mixed-integer programming problem exactly can be difficult in practice for consolidation trucking networks of larger size. To understand the likely size of the optimization problems, consider an LTL carrier operating in North America with 100 terminals. If freight demand exists between half of the origin-destination terminal pairs (which is likely an underestimate), the result is a model with roughly 5000 commodities. Suppose further that 30 terminals are hubs and 70 are smaller end-of-lines; then, we should expect at least roughly 900 directed arcs between hub pairs that might be used by any commodity, and an additional few hundred arcs connecting out of or into end-of-lines that can be used only by commodities originating or destined for those terminals. For a network of this size, the number of commodity flow variables x_{ij}^k is large. If each directed arc connecting two hubs serves all 5000 commodities, and if each directed arc connected to an end-of-line serves approximately 50 inbound or outbound commodities, the model may have more than 4 million commodity flow variables and over 150,000 flow balance constraints of type (14.9). Thus, the linear relaxation of (14.8)–(14.13) is a very large linear program. Integer load counting variables τ_{ij} are defined for each arc, so there are at least 1000 of these variables and perhaps more. It can be important in practice to attempt to limit the number of x_{ij}^k variables by restricting which commodities might ever use specific hubs.

To date, exact approaches for solving these problems rely on using cutting planes to strengthen the mixed-integer programming formulation; results have been reported for instances with up to 100 nodes, 400 arcs, and 200 commodities, still too small for application to many real-world consolidation trucking networks. A simple yet effective cutting-plane algorithm for the flow planning MCND problem works as follows. Define *strong inequalities* for all commodities k and lanes (i, j) by:

$$x_{ij}^k \leq q_k \tau_{ij}. \tag{14.14}$$

The strong inequalities are clearly valid, and it has been shown that they are facet-defining for the convex hull of the so-called single-arc design relaxation of the MCND. Intuitively, the benefit of the strong inequalities should be clear when each individual commodity demand is small compared to the capacity of a single trailer Q: we can interpret the inequality as forcing the solution to allocate at least a partial single trailer on any lane where commodity k moves, and at least one trailer if all commodity k flow moves on the lane. Although strong inequalities are helpful, it is likely impractical to introduce them all to the formulation given the number of commodity flow variables. Separation of these inequalities, however, is simple because they can be checked directly for each lane and commodity. Thus, a reasonable exact solution approach for the flow planning MCND problem is to solve the linear programming relaxation at the root, and then to iterate introducing violated strong inequalities and resolving until no violations remain. The resulting linear programming formulation, extended with the identified subset of strong inequalities, is then solved by reintroducing the integrality constraints (14.13) and calling a MIP solver.

3.2 Single-Path and In-Tree Flow Planning Models

In addition to being difficult to solve in its generic form, the flow planning model (14.8)–(14.13) also has a number of drawbacks that limit its usefulness in practice. One deficiency is that flow for each commodity k can be split across potentially many paths connecting o_k to d_k, and the fraction of commodity moved on any such path may be arbitrarily small. When developing a flow *plan* for consolidation, LTL and package trucking carriers seek both realism and simplicity. During operations, the actual freight shipments (and total flow volume) for any commodity may differ from the expected flow used for planning so it is at best not clear when to choose one path for a particular shipment versus another. While we believe that it is appropriate that such models ignore the details of individual shipments and model demand as continuous commodity flows, it is at the same time likely necessary to exercise some control over flow splitting during this planning phase. Of course, in practice carriers may divert shipments during operations onto alternative transfer paths through different cross-dock terminals.

A simple but restrictive way to eliminate commodity flow splitting is to enforce a *single path* constraint when building a flow planning model. Here, all commodity k flow is directed to a single transfer path from o_k to d_k in the plan. Consider the following formulation that embeds this restriction. Suppose we also introduce a new mechanism to model commodity flow where variables y_{ij}^k now measure the *fraction* of commodity k demand volume that is transferred directly from terminal i to j. Using this redefinition, $x_{ij}^k = q_k y_{ij}^k$. Now, if we restrict the y variables to be binary, we can easily enforce a single-path restriction:

$$\text{minimize} \quad \sum_{(i,j)\in A} (d_{ij}\tau_{ij} + c_{ij}x_{ij}) \tag{14.15}$$

subject to

$$\sum_{(i,j)\in A} y_{ij}^k - \sum_{(j,i)\in A} y_{ji}^k = \begin{cases} 1 & \text{if } i = o_k \\ -1 & \text{if } i = d_k \\ 0 & \text{otherwise} \end{cases} \qquad \forall k \in K,\ \forall i \in N \tag{14.16}$$

$$x_{ij} \le Q\tau_{ij} \qquad \forall (i,j) \in A \tag{14.17}$$

$$x_{ij} = \sum_{k\in K} q_k y_{ij}^k \qquad \forall (i,j) \in A \tag{14.18}$$

$$y_{ij}^k \in \{0,\ 1\} \qquad \forall k \in K,\ \forall (i,j) \in A \tag{14.19}$$

$$\tau_{ij} \ge 0 \text{ and integer} \qquad \forall (i,j) \in A \tag{14.20}$$

Constraints (14.18) convert commodity flows into total flow on direct lanes, and thus the objective function and trailer counting constraints can remain as in the initial model. Constraints 14.16 ensure that all commodity k demand is transferred from o^k to d^k. Furthermore, when the y variables are restricted to take binary values, these constraints ensure that the y^k variables identify a single path from o_k to d_k. It is again possible to introduce strong inequalities to this formulation of the form $y_{ij}^k \le \tau_{ij}$ to strengthen the linear programming relaxation.

This single-path flow planning formulation can be referred to as an *unsplittable flow* capacitated network design problem (see e.g., Atamtürk and Rajan, 2002). Note that the original splittable formulation can also be modeled with y variables by relaxing (14.19) to $y_{ij}^k \in [0, 1]$; the original formulation can be recovered by substituting $x_{ij}^k = q_k y_{ij}^k$ in this case.

Practical models for truck flow planning are often even more restricted. Consider the sort operations at a trucking terminal i. Arriving trucks are unloaded, and shipments that must be transferred (since they have not arrived at their final destination) are sorted for loading into outbound trailers. In typical operations, each outbound trailer is destined for a single next terminal j where it will be unloaded entirely. Technology is certainly available today for each shipment to have a customized sorting plan; at terminals with appropriate technology, a shipment can be scanned and then sorted for loading onto an appropriate outbound trailer (by a terminal worker or by automated sorting equipment). However, carriers often operate simpler plans that specify rules that guide how groups of shipments are to be sorted. One option still used in many LTL and package express systems is to determine the next terminal for each unloaded shipment using only its final destination.

Suppose that a consolidation plan is such that all freight shipments unloaded at terminal i with the same final destination d ($i \neq d$) are transferred to a single next terminal j. We will call such a design an *in-tree flow plan* because the directed graph induced by the union of paths for commodities $K_d \subseteq K$ that share a common destination d is a directed in-tree on (a subset of) the terminal nodes N.

In-tree plans are a subset of the feasible single-path plans, and we can modify the formulation to handle this restriction. To do so with the simplest formulation, we introduce and make use of a common redefinition of commodities that is frequently used in network design. It is well known that some multi-commodity network design problems can be formulated where each commodity represents all shipment flow to a common destination (or alternately, all shipment flow from a common origin); these redefined commodities are referred to as *aggregated* commodities. In the destination variant of aggregated commodities, let D be the set of destinations d that have positive inbound freight flow. If we begin with our original origin-destination commodity definition, then we can define aggregated commodities for each destination d with the following net supply of flow:

$$b_i^d = \begin{cases} q_k & \text{if } i = o_k \text{ for some } k \in K_d \\ -\sum_{k \in K_d} q_k & \text{if } i = d \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, d \in D,\ \forall\, i \in N. \tag{14.21}$$

It is descriptive to refer to this type of aggregation as a many-to-one commodity into terminal d. Aggregated commodities have the obvious benefit of reducing problem size. If the number of terminals $|N| = n$, then the number of possible flow decisions for each arc is $O(n)$ instead of $O(n^2)$ and the number of flow balance constraints is also reduced by $O(n)$. However, aggregated commodities may not be useful when there is the need to explicitly model specific origin-destination flow path requirements, for example path duration or cardinality constraints. It is also not possible to use aggregated commodities to model a single-path flow planning problem unless the paths inbound to every destination d are constrained also to form a directed in-tree.

Consider now an in-tree flow planning model with many-to-one commodities. Let binary decision variable y_{ij}^d be used to indicate whether freight flow for commodity d (with final destination d) that originates or transfers at terminal i is transferred next to terminal j. Let continuous variable x_{ij}^d measure freight flow for commodity d moving on truck trailers from i to j, as usual. Consider the following formulation:

$$\text{minimize} \quad \sum_{(i,j)\in A} (d_{ij}\tau_{ij} + c_{ij}x_{ij}) \tag{14.22}$$

subject to

$$\sum_{(i,j)\in A} x_{ij}^d - \sum_{(j,i)\in A} x_{ji}^d = b_i^d \qquad \forall d \in D,\ \forall i \in N \tag{14.23}$$

$$\sum_{(i,j)\in A} y_{ij}^d \le 1 \qquad \forall d \in D,\ \forall i \in N \tag{14.24}$$

$$x_{ij}^d \le \left(\sum_{k\in K_d} q_k\right) y_{ij}^d \qquad \forall d \in D,\ \forall (i,j) \in A \tag{14.25}$$

$$x_{ij} \le Q\tau_{ij} \qquad \forall (i,j) \in A \tag{14.26}$$

$$x_{ij} = \sum_{d\in D} x_{ij}^d \qquad \forall (i,j) \in A \tag{14.27}$$

$$y_{ij}^d \in \{0,\ 1\} \qquad \forall d \in D,\ \forall (i,j) \in A \tag{14.28}$$

$$\tau_{ij} \ge 0 \text{ and integer} \qquad \forall (i,j) \in A \tag{14.29}$$

The in-tree model includes constraint (14.24) to ensure that the freight flow paths for destination d form a directed in-tree to d by allowing only one outbound direct transfer arc (i, j) to be selected from terminal i for that freight. Constraint (14.25) ensures that freight destined for d can only be dispatched on lanes included in the selected in-tree, where the capacity coefficient for y_{ij}^d is the smallest big-M value that yields a valid formulation. Again, valid inequalities can be introduced to this formulation. For example, replacing y_{ij}^d with τ_{ij} in (14.25) yields a version of the strong inequalities. Inequalities $y_{ij}^d \le \tau_{ij}$ are also valid for all $d \in D$ and $(i, j) \in A$.

3.3 Path-Based Models for Flow Planning

Each of the flow planning formulations presented thus far has been an *arc-based* network design model. For parsimony, such models use arc flow decision variables of the form x_a^k and x_a to represent respectively the fractional freight flow for commodity k and all commodities moved via truck dispatch arc a. When many possible feasible paths exist for routing commodity k freight from o_k to d_k, this modeling decision has merit since it may reduce the number of required decision variables (and this remains true even for the single-path and in-tree models that select only a single path for each such pair).

In many truck transportation applications, however, it is better to use a *path-based* network design model because of the flexibility these models provide in representing specific restrictions that may arise in practice. Such models replace the variables x_a^k with path-flow variables x_p^k, where p represents some path in P_k for commodity k. We can then modify the generic arc-based model into the following generic path-based model:

$$\text{minimize} \quad \sum_{(i,j)\in A} (d_{ij}\tau_{ij} + c_{ij}x_{ij}) \tag{14.30}$$

subject to

$$\sum_{p\in P_k} x_p^k = q_k \qquad \forall\, k \in K \tag{14.31}$$

$$x_{ij} \le Q\tau_{ij} \qquad \forall\, (i,j) \in A \tag{14.32}$$

$$x_{ij} = \sum_{k\in K} \sum_{p\in P_k \,|\, (i,j)\in p} x_p^k \qquad \forall\, (i,j) \in A \tag{14.33}$$

$$x_p^k \ge 0 \qquad \forall\, k \in K,\ \forall\, p \in P_k \tag{14.34}$$

$$\tau_{ij} \ge 0 \text{ and integer} \qquad \forall\, (i,j) \in A \tag{14.35}$$

Since all paths in P_k provide connectivity from o_k to d_k, flow balance constraints are no longer required and instead are replaced by (14.31) which partitions commodity demand flow across the available paths in P_k. Constraints (14.33) aggregate all commodity flow on direct movement arc (i, j) by finding all flow on paths for all commodities where the arc (i, j) is included in the path. Note also that although (14.30) does not include a linear cost term for the path flow variables, adding one is possible; in practice, such terms can be used to model freight handling costs at intermediate cross-dock transfer terminals rather than using a cost linear in the total arc flow x_{ij}.

The primary benefit of such a formulation is that the model explicitly defines the sets of allowed transfer paths P_k for each commodity. This feature makes it easy to model a number of real-world restrictions. Most importantly, suppose that a service requirement requires that the duration of the transfer path for commodity k from o_k to d_k (travel time plus terminal cross-dock time) is limited by an upper bound. Then, only paths that meet this duration requirement can be included in P_k. Similarly, it may also be desirable to limit the number of terminal transfers for commodity k. For example, when o_k and d_k are nearby perhaps only one transfer should be considered in any path, but for more distant terminals two or three transfers might be acceptable; again, only acceptable paths need be included in P_k.

Building path sets P_k given o_k and d_k is usually conducted by using a graph search algorithm, like breadth-first or depth-first search, from o_k using the direct arcs $(i, j) \in A$. Constraints on allowable paths can be used to truncate the search tree. For many problems, enumerating the complete feasible path set P_k for each commodity k would create too many decision variables and very large instances. Column generation approaches for solving linear programming relaxations (either at the root node of a branch-and-bound tree or at all nodes in a branch-and-price scheme) can be used in these cases. Heuristics that only enumerate reasonably-sized subsets of the path pools for each commodity may also find good solutions in practice.

To use a path-based model while enforcing an in-tree flow plan structure, we can again add binary arc selection variables y_{ij}^d to the formulation and selection constraints (14.24). Suppose furthermore that the commodity set K is partitioned into subsets $K(d)$, where commodity k is included in $K(d)$ if its freight destination $d_k = d$. To ensure that the set of all paths used for commodities in $K(d)$ forms a directed in-tree into d, the following compatibility constraints can be used:

$$\sum_{p \in P(k) \mid a \in p} x_p^k \leq q_k y_a^{d_k} \quad \forall k \in K, \; a \in A \tag{14.36}$$

Note that these aggregated forcing constraints are stronger than those disaggregated by path, yielding a stronger linear relaxation formulation. The disaggregated constraints have the simpler form $x_p^k \leq q_k y_a^{d_k}$ and are defined for all k, $p \in P(k)$, and $a \in p$; it is easy to see that there are feasible solutions to the disaggregated constraints system when the variables are continuous that are not feasible for the aggregated constraint. Of course, in-tree constraints will force all commodity k flow onto a single path $p \in P(k)$ into d_k for integer values of y. It is thus possible to redefine x_p^k in this case to be a binary selection variable, and to modify constraints (14.31) into assignment constraints with right-hand side values of one. After this modification, the aggregated forcing constraints take the form $\sum_{p \in P(k) \mid a \in p} x_p^k \leq y_a^{d_k}$ for all k and a. Finally, there are some terminals in consolidation trucking systems that cannot be used to transfer inbound freight from other terminals; most end-of-line terminals in LTL systems operate this way. When binary path selection variables are used, it is not necessary to include tree selection variables y_a^d for arcs a departing such terminals since originating flow destined for d will automatically be forced onto a single path and no transfer freight exists.

3.4 Balancing Resources in Flow Planning

The model of transportation costs discussed in Sect. 3.1 assumes that trailer movements τ_{ij} and their associated costs are determined only by one-way loaded flows and thus ignores the important fact that trailer and container resources are reused over time. Empty trailers must be available at load origins before loads can be created and moved, and the problem of *empty repositioning* of equipment is critically important in most freight transportation settings. When empty repositioning is ignored during flow planning, opportunities to use empty trailer capacity to move freight may be overlooked. Flow planning models that explicitly include empty resource flows and their associated costs seek to address this shortcoming.

To show how planned flow costs may be reduced by integrating empty repositioning decisions in flow planning when compared to the sequential deployment of a flow planning model followed by an empty trailer balancing problem, consider a simple example with 3 terminals as illustrated in Figs. 14.2 and 14.3. Suppose $\frac{1}{2}$ trailerloads of demand exists from a to b and from a to c and one trailerload from b to c. Suppose that the distance from a to b or c is $\frac{2}{3}$ and the distance from b to c is one. In the absence of empty balance, the optimal solution is to move full trailerloads on lanes (a, b), (a, c), and (b, c); doing so creates imbalance and two trailers should be returned on leg (c, a). The total trailer distance in this solution is $\frac{11}{3}$, but the loaded trailer distance is only $\frac{7}{3}$. If empties were balanced simultaneously, it is better to load the a to b freight via terminal c. This creates loaded trailers on (a, c), (b, c), and (c, b) and total loaded trailer distance of $\frac{8}{3}$. However, empty balance can be achieved by only sending one empty on (c, a) and thus the total trailer distance is $\frac{10}{3}$.

Up to this point, the set A of arcs $a = (i, j)$ has been used to represent opportunities to move loaded trailers from terminal i to terminal j; note that loading of the trailer occurs at terminal i and unloading at terminal j. It is certainly possible to limit empty trailer movements to the arcs in A. There are some cases, however, when empty trailers might move between terminals where there is *never*

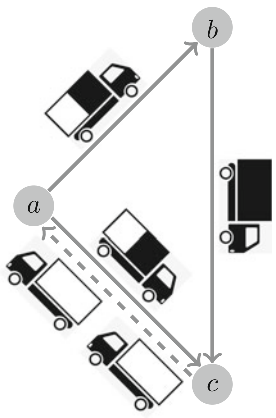

Fig. 14.2 Flow plan without empty balancing (Loaded distance = $\frac{7}{3}$, totaltrailer distance = $\frac{11}{3}$)

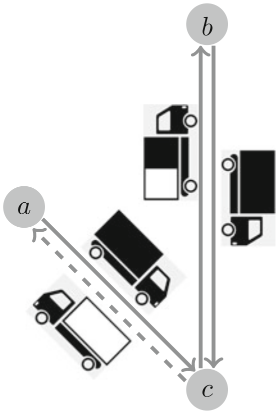

Fig. 14.3 Flow plan with empty balancing (Loaded distance = $\frac{8}{3}$, totaltrailer distance = $\frac{10}{3}$)

a trailer loaded at i to be unloaded at j. Examples that have been encountered in practice include pairs of hubs in large metropolitan areas, nearby pairs of end-of-line terminals, or connections between large customer facilities (another type of end-of-line). To model terminal-to-terminal physical movements of trailers more explicitly, let A^D be a set of *dispatch lanes* for a trucking company that includes A but may also include additional connections where empties may be moved: $A \subseteq A_D$. It is also common in practice for A_D to have an important property: a directed path of dispatch lanes in A^D should exist from j to i, for each $(i, j) \in A$. The existence of the reverse path is a sufficient (but not necessary) condition to ensure that empty resources can return to load origins for reuse. If the reverse path does not exist for each $(i, j) \in A$, which is unlikely in practice, then it is important to guarantee that empty trailers can be balanced using a different mechanism.

Suppose now that empty trailers and containers can be moved on any dispatch lane $(i, j) \in A^D$. Let η_{ij} count the number of empty unit loads moving on dispatch lane (i, j), measured in the same units as τ_{ij}. We can enforce equipment balance then at each terminal i by adding the following constraint to any of the flow planning formulations presented thus far:

$$\sum_{(i,j)\in A} \tau_{ij} + \sum_{(i,j)\in A^D} \eta_{ij} - \sum_{(j,i)\in A} \tau_{ji} - \sum_{(j,i)\in A^D} \eta_{ji} = 0 \quad \forall i \in N \tag{14.37}$$

Given this balance constraint, flow plans can be determined by including an appropriate cost for moving empty trailers in the objective function. If d_{ij}^E is the cost of moving an empty unit load on dispatch lane (i, j), then we can add the term $\sum_{(i,j)\in A^D} d_{ij}^E \eta_{ij}$ to any of the objective functions to capture empty costs. Doing so is likely to lead to changes in the optimal flow plans and empty trailer balance plans that would result from solving the problems sequentially: some freight will be assigned optimally into natural empty backhaul corridors, and balancing backhaul trailer movements may deviate from the most direct (cheapest) paths to attract freight.

3.5 Slope-Scaling Heuristics for Flow Planning

Flow planning optimization models are important in practice for the design of consolidation transportation networks, but exact optimization can be difficult for instances of realistic size. One heuristic approach to solving these problems is to simply limit the number of decision variables that are defined to a tractable number; this idea is easiest to implement with path-based models as we described earlier, but it is also possible with arc-based approaches. Slope scaling is a different heuristic idea and can be useful for problems that have difficult non-linear objective functions that, when linearized, lead to optimization problems that can be solved efficiently. We will now describe how slope scaling can be used to solve flow planning problems, and how the linearized subproblems decompose into shortest-path problems for both the base model and the single-path model (whose slope-scaling solutions are therefore equivalent). Furthermore, the intree model can also be solved by a shortest-path decomposition (and is therefore equivalent to the base and single-path models) when the objective function cost coefficients are independent of commodity k.

Slope scaling heuristics are useful for problems with difficult non-linear objective functions but with linear constraints and decision variables. The generic model (14.1)–(14.5) has this form, since its constraints are separable by commodity k and, when separated, describe the minimum-cost path polytope since all commodity k flow has a single origin (as well as a single destination) and arc flows are uncapacitated. We now show how to use slope scaling to find solutions first for model (14.8)-(14.13). Note that we can eliminate the integer dispatch variables τ_{ij} by rewriting the objective function recognizing that cost-minimizing values for τ_{ij} follow directly from the flow variables since it is non-decreasing in τ. Thus, we can define:

$$\tau_{ij} = \left\lceil \frac{\sum_{k \in K} x_{ij}^k}{Q} \right\rceil, \tag{14.38}$$

and the optimization model can be rewritten as minimizing the objective function:

$$\sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ij}^k + \sum_{(i,j) \in A} d_{ij} \left\lceil \frac{\sum_{k \in K} x_{ij}^k}{Q} \right\rceil - \sum_{i \in N} \sum_{k \in K \mid o_k = i} h_i q_k \tag{14.39}$$

subject to (14.9) and (14.12).

To solve with a slope-scaling approach, we linearize the objective function by replacing the ceiling function:

$$\sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ij}^k + \sum_{(i,j) \in A} \rho_{ij}(t) \sum_{k \in K} x_{ij}^k - \sum_{i \in N} \sum_{k \in K \mid o_k = i} h_i q_k, \tag{14.40}$$

where $\rho_{ij}(t)$ is the slope coefficient in iteration t. Rearranging terms yields:

$$\sum_{k\in K}\sum_{(i,j)\in A}\left(c_{ij}+\rho_{ij}(t)\right)x_{ij}^k-\sum_{i\in N}\sum_{k\in K\,|\,o_k=i}h_i q_k, \tag{14.41}$$

A slope scaling heuristic finds each solution to the flow planning problem by selecting a fixed vector $\rho_{ij}(t)$, then determining x_{ij}^k that minimize (14.41) subject to (14.9) and (14.12), and then finally specifying trailer flow variables using (14.38). It should be clear that the minimization problem for $\rho_{ij}(t)$ is a linear program that is separable by commodity k. Moreover, each separable subproblem is to find a minimum-cost path from o_k to d_k for commodity k given arc cost coefficients $c_{ij}+\rho_{ij}(t)$. Since this solution is a single flow path for each commodity k, it also follows that the solution to any slope scaling subproblem will also be a single-path flow plan; thus, solving (14.8)–(14.13) is equivalent to solving (14.15)–(14.20) by slope scaling. Furthermore, note also that the objective coefficients on commodity arc flow in (14.41) are independent of commodity k. If we consider all commodities k that share a common destination $d_k=d$, we can find a joint set of minimum-cost paths given arc costs $c_{ij}+\rho_{ij}(t)$ using an algorithm that produces an in-tree to destination d, like Dijkstra's Algorithm. Thus, a solution found during any slope-scaling iteration for an in-tree flow planning problem is also optimal for the base or single-path problems with the same linearization multipliers $\rho_{ij}(t)$. If instead the commodity arc flow cost coefficients had the more general form $c_{ij}^k+\rho_{ij}(t)$, separate shortest path problems would be necessary for each commodity for the base and single-path slope scaling problems. Furthermore, the slope scaling problem for the in-tree flow planning model would require solving mixed-integer program for each destination d to enforce the tree structure on the joint set of paths.

Consider then the following slope scaling approach for solving the base, single-path, or in-tree flow planning problem. We initialize the slope coefficients using a lower-bounding approximation: $\rho_{ij}(1)=\frac{d_{ij}}{Q}$ for each arc. Then, for each iteration t, we minimize (14.41) subject to (14.9) and (14.12) by first finding a shortest-path in-tree to each destination d using arc costs $c_{ij}+\rho_{ij}(t)$, then assigning commodity flow q_k along the identified shortest path from o_k to d_k for each $k\in K$ yielding $x_{ij}^k(t)$, and finally determining arc trailer flows $\tau_{ij}(t)$ using (14.38). The true objective function cost of this solution is $C(t)$, determined using (14.8). If $C(t)$ is the lowest objective function cost found so far, it is recorded and the best solution is updated. If the solution $x_{ij}^k(t)$ remains unchanged from the prior iteration $t-1$, then we terminate and return the best found solution. Otherwise, we adjust the slope coefficients as follows and move on to iteration $t+1$:

$$\rho_{ij}(t+1)=\begin{cases}\rho_{ij}(t) & \text{if } \sum_{k\in K}x_{ij}^k(t)=0\\ \dfrac{d_{ij}\left\lceil\frac{\sum_{k\in K}x_{ij}^k(t)}{Q}\right\rceil}{\sum_{k\in K}x_{ij}^k(t)} & \text{if } \sum_{k\in K}x_{ij}^k(t)>0\end{cases} \tag{14.42}$$

Note finally that it may also be reasonable to terminate a slope-scaling search after a maximum number of iterations to avoid excessive computation time.

3.6 A Local Search Heuristic for Flow Planning

Local search and metaheuristic extensions of local search are important heuristic approaches for solving flow planning problems. Most heuristics of this type take advantage of the fact that minimum cost path algorithms can frequently be used, with modifications to arc costs or network structure, to decide on new freight flow paths for commodities during search iterations. We now describe the core ideas of a local search approach that for many years was used as a key component in linehaul network design for LTL carriers; see Sect. 5 for more background. The ideas presented here will closely follow those developed initially in Powell (1986) for what was then referred to as the load planning problem for LTL carriers; in the terminology of this chapter, the problem considered is a flow planning problem.

The base problem considered is to create an in-tree flow plan of the type described by constraints (14.23)–(14.28), however we will assume only non-negative trailer flow τ_{ij} as described below. The LTL flow planning problem will be to decide τ_{ij} on direct movement lanes $(i, j) \in A$ and freight flows x_{ij}^d for each aggregated destination commodity $d \in D$. However, this planning problem considers a simpler transportation cost function $f_{ij}^T(x_{ij})$ given total freight flow x_{ij} on lane (i, j) measured in fractional trailerloads. If any flow is assigned to the lane, we incur a fixed cost equivalent to dispatching a minimum flow of trailers M_{ij} at cost $d_{ij}M_{ij}$. Once the capacity of this minimum flow is exceeded, we approximate additional trailer dispatching cost with a linear term $d_{ij}x_{ij}$. If we use (14.22) as the flow planning objective function, then we represent this cost approximation by determining the trailer flows τ_{ij} as follows:

$$\tau_{ij} = \begin{cases} 0 & \text{if } x_{ij} = 0 \\ M_{ij} & \text{if } 0 < x_{ij} \leq M_{ij} \\ x_{ij} & \text{if } M_{ij} < x_{ij} \end{cases} \tag{14.43}$$

Note that the objective function cost term $d_{ij}\tau_{ij}$ exhibits cost scale economies in x_{ij} for the first M_{ij} units of freight flow to encourage consolidation. If M_{ij} were one for all (i, j), this trailer counting function is a lower bound for the trailer step function introduced earlier in model (14.8)–(14.12). If q_k measures weekly rates of demand, then $M_{ij} = 5$ would indicate that at least one trailer should be sent each weekday on dispatch lane (i, j) if any flow is assigned to that lane. Minimum dispatch frequencies can be used to ensure that reasonable service levels are provided to commodities that use this lane in their (o_k, d_k) path.

We now describe a two-tier local search heuristic for finding solutions to this model. The first tier heuristic selects a subset $A_S \subseteq A$ of direct service lanes (i, j) to make available for use; including lane (i, j) in A_S implies that a minimum of M_{ij} trailers will flow on the lane. A feasible first-tier solution is one where at least one freight path exists between o_k and d_k for each (disaggregated) commodity $k \in K$ using only arcs in A_S. Given such a feasible A_S, the second tier heuristic determines a freight flow path for each $k \in K$ to minimize the value of the objective function. The flow q_k for commodity k is assigned to each arc (i, j) in its path, thus determining the total arc flows x_{ij} and trailer flows t_{ij} and the concomitant objective function value. Note that the joint set of freight flow paths into each $d \in D$ must form a directed in-tree.

We begin with the second tier problem, referred to in the literature as the *routing subproblem*. First, note that this minimization problem is on its own a piecewise-linear convex multi-commodity flow problem with a relatively simple cost structure. If we rewrite the objective function as

$$\min \sum_{(i,j)\in A_S} (D_{ij} + c_{ij}x_{ij}), \tag{14.44}$$

we can replace the trailer flow variables with D_{ij} by adding the following constraints:

$$D_{ij} \geq d_{ij}M_{ij} \qquad \forall (i, j) \in A_S \tag{14.45}$$

$$D_{ij} \geq d_{ij}x_{ij} \qquad \forall (i, j) \in A_S \tag{14.46}$$

and we can solve a mixed-integer programming problem with constraints (14.23)–(14.25) and (14.27) and (14.28). Note that the in-tree structure is what makes this optimization problem difficult since it is otherwise a linear program. A reasonable solution approach might be to only introduce tree selection variables and constraints for nodes i and destinations d when violations occur when they are ignored.

Another idea is to solve the second tier routing subproblem by local search. To do so, note that a feasible set of in-tree paths for each destination d can be found by solving a shortest-path problem with arc costs $d_{ij} + c_{ij}$. By moving all q_k flow for commodity k along this shortest path, it should also be clear that the total cost of the resulting arc flows $\overline{x}_{ij}$,

$$\sum_{(i,j)\in A_S} \left(d_{ij} + c_{ij}\right)\overline{x}_{ij}, \tag{14.47}$$

is a lower bound on the objective function value for the routing subproblem. In fact, if $\overline{x}_{ij} \geq M_{ij}$ for all arcs, this solution is optimal. Thus, the only way to improve a solution is to find arcs where $\overline{x}_{ij} < M_{ij}$ and determine whether any commodities can be re-routed to use them and reduce cost. To do so, let an *override* indicate when a specific value y_{ij}^d is forced equal to one by the heuristic; the goal of setting an override will be to force a shortest-path algorithm to include arc (i, j) surely in the in-tree to destination d. Given a set of overrides, it is easy to modify an algorithm, like Dijkstra's, to find a shortest-path in-tree to d conditional on including all arcs where $y_{ij}^d = 1$. The idea is simple: when extending labels from j to upstream nodes k along arcs (k, j), we only update the cost label (dual) at k if its cost improves and either $y_{kj}^d = 1$ or no override is set outbound from node k.

Thus, a core component of a heuristic for the routing subproblem will be to determine commodity flows $\overline{x}_{ij}^d$ and arc flows $\overline{x}_{ij}$ by building shortest-path in-trees to each destination $d \in D$ modified by current overrides y_{ij}^d. Given this solution, we can seek to reduce its cost by attracting flow to arcs where $\overline{x}_{ij} < M_{ij}$. Let $\overline{y}^d$ indicate the current in-trees selected for this solution. One approach to attracting flow to (i, j), described originally in Powell (1986) as IFOL-0, is to consider all destinations $s \in D$ where $\overline{y}_{ij}^s = 0$ and some flow quantity $q^s(i)$ has accumulated at i for transfer onward to s. Note that $q^s(i)$ may include both originating flow at i and also the sum of flows from upstream origin terminals k such that the path from k to s on the tree arcs a where $\overline{y}_a^s = 1$ includes terminal i. If we were to change the in-tree for s such that (i, j) were included in the in-tree (by setting override $y_{ij}^s = 1$), then the new path for this flow would include arc (i, j) and then follow the tree arcs from j to s. In the original IFOL-0 approach, the destinations s are sorted by decreasing estimated savings from diversion onto (i, j) computed by $q^s(i)$ multiplied by the *marginal* cost of the new path (where an arc marginal cost is zero if its flow is less than M_a, and c_a otherwise), and then processed in order of estimated savings where commodity s is diverted onto (i, j) only if it generates actual cost savings.

Given an approach for the second tier routing subproblem, it remains to discuss a heuristic approach for selecting the arc subset A_S. This first tier problem, or network design master problem, can be addressed with a simple local search heuristic. Consider an LTL trucking network that currently dispatches loads on a set of lanes $\overline{A}_S$. Reasonable local search neighborhoods modify $\overline{A}_S$ by dropping a single arc or adding a single arc each iteration, generating a new routing subproblem solution and moving to the new solution generated only if total cost is reduced. When dropping an arc $a = (i, j)$, a set of destinations D_a where $y_a^s = 1$ for $s \in D_a$ is disrupted and a new in-tree needs to be constructed for each s to create a feasible solution. On the other hand, adding an arc a creates an opportunity to re-route flow to reduce cost and a procedure like IFOL-0 can be used to see which flow should be attracted to a. One approach to structuring such a heuristic would be to consider dropping all arcs from $\overline{A}_S$ first one-by-one, focusing first on those connecting breakbulk terminals to end-of-lines and vice versa and then moving to those connecting two breakbulks. After considering all such drops, a set of arcs could be considered for adding to the network. Several passes over a drop-add sequence should be conducted.

4 Network Design Models for Flow and Load Planning

Load planning in consolidation trucking is a more detailed task than flow planning and is at its core a scheduling activity. Given a flow plan, a trucking company needs to provide adequate transportation capacity between terminal pairs to support the flows over time. Generally a *schedule* is constructed for a time horizon, like a week or a month, to provide this capacity. This schedule typically includes *loads*, *empties*, *movements or dispatches*, and *drivers*. A load is planned to be built at some origin terminal at some time, and then dispatched to a destination terminal to arrive by some time. A load also specifies a planned type of trailer or container equipment to be used (and its size). Empty loads (or empties) are planned both to recirculate equipment back to load origins but also, in some cases, to move drivers back to their home terminals. Movements, or dispatches, refer to terminal-to-terminal movements by drivers or outsourced transportation modes with one or more loads. The movement and driver schedule required to execute planned loads and empties is typically not considered part of the load planning problem.

Rate-based flat network service network design models are still useful for flow planning, and path-based variants in particular can model some important timing considerations for commodities. However, they have a few important drawbacks. Flat network models are not particularly useful for detailed flow and load planning primarily because they do not accurately model the *timing* of consolidation activities at transfer terminals or details about when equipment is available for dispatch. For this reason, flat network models are not usually deployed for *load planning* problems that seek to explicitly create plans for timed dispatches of trucks during an operating day. We now introduce *time-expanded* network models for such service network design problems. In this section, we will use the term *flow and load planning models* to refer to those that both create capacity and plan shipment flows through a consolidation network while simultaneously planning loaded and empty trailer dispatches during a planning horizon.

4.1 A Time-Expanded Model for LTL Flow Planning

Before we explore models for joint flow and load planning, we introduce an important time-expanded network model for LTL flow planning; the model and solution approach in this section were first described in Jarrah et al. (2009). Consider a time-expanded network where $\mathscr{N}$ and $\mathscr{A}$ represent the set of time-space (terminal, time) nodes and time-space arcs (denoting the timed trailer dispatch lanes), respectively. To model the network over time, suppose that each geographic terminal in N is replicated once for each of the five weekdays to yield the nodes in $\mathscr{N}$. Similarly, each geographic load dispatch lane in A is also replicated for each weekday to yield the arcs in $\mathscr{A}$. Note that load lanes that take more than a single travel day to reach their destination are connected forward to the appropriate destination node in $\mathscr{N}$. The arcs in $\mathscr{A}$ also now include holding arcs forward one time period (weekday) for each terminal node in $\mathscr{N}$. Since carriers actually dispatch loads at more than a single time per day, this model is best described as one of tactical flow planning rather than a detailed load planning and dispatch model.

Given this time-expanded network structure, each commodity demand now specifies a timed origin node and timed destination node, both in $\mathscr{N}$. Using this structure, the volume of freight moving between geographic terminals can be modeled to vary by day-of-week. Furthermore, the transit time requirement for each origin-destination pair can be modeled (at the level of days) by choosing an appropriate timed destination node.

If the goal is to produce an in-tree flow plan for each terminal destination $d \in N$, it is possible to formulate the flow planning problem with binary in-tree selection variables. To do so, it is possible to modify a path-based flow planning model like (14.30)–(14.36) to one where the primary binary decision variables are w_ℓ^d, indicating whether or not complete in-tree ℓ is selected for destination $d \in N$. This approach assumes that the same in-trees will be used each operating day of the week in the flow plan. In such a formulation, each in-tree is comprised of time-expanded paths from origin terminals into d, replicated for each operating day and consistent with the in-tree property that only a single outbound terminal $j \in N$ can be selected for flow outbound from $i \in N$ for destination d. In fact, since the model also includes holding arcs, the in-tree property is extended in this case to holding: if any freight at (i, t) destined for d is held to $(i, t + 1)$, then all such freight must be held.

A feature of such an in-tree model is that a unique time-expanded path connects each timed origin to a timed copy of d. Thus, if a specific tree ℓ is selected for d, a precise mapping of commodity freight volumes destined to d to time-expanded arcs is known; in this way, a set of tree selection decisions implies freight volumes on all time-expanded arcs which in turn specifies load counts and fixed transportation costs. Additionally, empty trailer balancing constraints are included in this model and are a straightforward extension of constraints (14.37) to the case with time-expanded nodes and arcs.

Specifying a flow planning model with tree variables is convenient, but the drawback in practice is that there are far too many feasible in-trees for each destination to enumerate. A heuristic approach to solve this integer programming model is to use a slope-scaling heuristic to linearize the fixed costs, and then to use column generation to solve the resulting linear programs without enumerating all feasible in-trees. The slope-scaling approach proposed here is very similar to the generic approach presented in Sect. 3.5.

Given a set of slope-scaling linearization factors, it can be shown that the empty balancing problem is independent of the tree selection variables. Thus, the slope-scaling linear programming problem can be decomposed into a simple linear subproblem for empty balancing (with fractional empties) and another for selecting in-trees for each destination. The empty problem needs only to be solved once. The in-tree selection linear subproblem is also simple, and in fact can be solved by inspection by choosing for each destination the tree with the smallest cost coefficient; thus, the in-tree selection problem results in integer solutions.

All of these observations motivate the following approach for solving the slope-scaling LP for a given set of linearization factors. Given an initial set of possible in-trees ℓ with at least one per destination, the in-tree selection LP is solved (by inspection). The resulting dual variables associated with the tree selection constraints are used by an integer program that is solved for each destination d that seeks to find (if possible) a new in-tree with negative reduced cost. This in-tree selection integer program will not be described in more detail here, but it should be noted that it is based on enumerating sets of possible time-expanded paths into the timed node copies of destination d in $\mathscr{N}$ and using binary variables to select a joint set that satisfies the in-tree property. Once the LP is solved to optimality via this column generation procedure, the slope scaling multipliers are updated using the approach described in this chapter and the LP is solved iteratively until a stopping criterion is met.

This time-expanded flow planning approach was implemented in practice for a major US LTL carrier. The carrier at the time operated nearly 150 terminals, and thus needed to solve large-scale instances with up to 725 time-expanded nodes, 30,000 arcs, and 680 time-space destination commodities. Computational results demonstrate that the algorithm was able to find improvements of 4–5% in flow plan costs when compared to the carrier's base flow plan.

4.2 Time-Expanded Models for LTL Flow and Load Planning

Modern LTL operators often provide services between many origin-destination terminal pairs with rapid transit times, often as short as overnight or 2 days. Even with such tight time constraints, it still may make sense to transfer freight multiple times at intermediate hubs. In such situations, the *timing* of consolidation is critical: will the cross-docking occur in the overnight hours, or during the day, or in the evening with freight picked up that day from the local operations? Models that attempt to determine flows and build a schedule of loads need detailed timing to make these decisions accurately.

Consider then a modeling framework for flow and load planning problems that includes the following features: (1) detailed time-space network modeling, where nodes denote (terminal, time) pairs and arcs denote timed movements, with fine time discretization (with multiple decision epochs in a day for each terminal) representing a single week of activity, (2) integrated consideration of loaded and empty trailer movements, and (3) support for flexible plans that use an in-tree flow plan structure but do not require the same trees or schedule of loads every day of the week.

We now specify a path-based flow and load planning model with these features. Each commodity k now specifies an origin and destination terminal as usual but additionally is associated with a specific day-of-week and a latest delivery day and time at the destination; since originating freight arrives primarily from the pickup-and-delivery operation, it is assumed that it all becomes available simultaneously in the evening of each day (for example, 7 p.m.). Paths for each such commodity are now sequences of load dispatches and terminal holding arcs that denote waiting at terminals. In-tree structure is enforced for terminal nodes regardless of the time-of-day of individual dispatches. It is not difficult to allow a different in-tree structure for destination d at different times during the planning horizon (for example, a new in-tree can be specified for each day-of-week separately). For simplicity of exposition, in this chapter we present the model where a single in-tree per destination terminal

d is specified that persists for the entire planning horizon; the extension where the tree arc choices can vary by day-of-week is not very different.

For the time-expanded network formulation, let $\mathscr{N}$ and $\mathscr{A}$ again represent the set of time-space (terminal, time) nodes and time-space arcs (denoting the timed trailer dispatch lanes), respectively. The set of time-space arcs also includes freight holding arcs between consecutive nodes at the same terminal. Note that since loads and empties are now planned at specific times (denoted with decision variables τ_a), this becomes a load planning model. Commodity demand q_k is measured in fractional trailers, and the set of time-space paths of commodity k is denoted with $\mathscr{P}(k)$. Consistently with the notation used throughout, N and A refer to the geographic terminal locations and the geographic direct lanes connecting terminals, respectively. Each commodity is required to follow a single time-space path from origin to destination. Since in-tree variables are defined on the geographic network, it could be possible to have multiple paths for the same commodity that satisfy the in-tree requirement but this is prevented with the single-path constraints. The translation function $l(a)$ maps a time-space dispatch arc $a \in \mathscr{A}$ to its direct geographic lane $l(a) \in A$. Finally, $\Delta^+(u)$ is the set of all direct lanes $(u, j) \in A$. Then, we have the following time-space formulation for the path-based flow and load planning problem, which we denote PFLP-TS:

$$\text{minimize} \quad \sum_{a \in A} d_a \tau_a + \sum_{k \in K} \sum_{p \in \mathscr{P}(k)} h_p q_k x_p^k \tag{14.48}$$

subject to

$$\sum_{p \in \mathscr{P}(k)} x_p^k = 1 \qquad \forall k \in K \tag{14.49}$$

$$\sum_{l \in \Delta^+(u)} y_l^d \leq 1 \qquad \forall u \in N, \ \ \forall d \in N \tag{14.50}$$

$$\sum_{p \in \mathscr{P}(k): a \in p} x_p^k \leq y_{l(a)}^{d^k} \qquad \forall k \in K, \forall a \in \mathscr{A} \tag{14.51}$$

$$\sum_{k \in K} \sum_{p \in \mathscr{P}(k): a \in p} q_k x_p^k \leq \tau_a \qquad \forall a \in \mathscr{A} \tag{14.52}$$

$$\sum_{a \in \delta^+(i)} \tau_a - \sum_{a \in \delta^-(i)} \tau_a = 0 \qquad \forall i \in \mathscr{N} \tag{14.53}$$

$$x_p^k \in \{0, 1\} \qquad \forall k \in K, \ \ \forall p \in \mathscr{P}(k) \tag{14.54}$$

$$y_l^d \in \{0, 1\} \qquad \forall d \in N, \ \ \forall l \in \Delta^+(u), u \in N \tag{14.55}$$

$$\tau_a \in \mathbb{Z}_+ \qquad \forall a \in \mathscr{A} \tag{14.56}$$

Solving formulation PFLP-TS exactly, or its extension when in-trees are allowed to differ over time, is generally not possible for planning instances typically found in practice. Smaller regional LTL carriers with just a handful of terminals may lead to instances that can be solved by modern integer programming software, especially if care is taken to manage the number of feasible time-space paths for each commodity included in the sets $\mathscr{P}(k)$.

Larger carriers with hundreds of terminals can easily lead to instances with 500,000 time-space arcs, 50,000 commodities, and millions of feasible time-space commodity paths. Rather than trying to solve these integer programs exactly, then, we instead present *integer-programming-based local search* heuristics for finding solutions. In these approaches, all neighbors in the local search neighborhood are identified by feasible solutions to a smaller integer program, and the search for an improving solution is performed by solving that integer program.

In local search, we begin with a feasible incumbent solution (in this case, a set of feasible decision variables x, y, and τ) and search for a neighboring solution (or simply *neighbor*) whose cost is less than the cost of the incumbent. If such a solution is found, it becomes the new incumbent and we continue the search. The search can be terminated when a certain number of iterations has been performed, a time limit has been reached, or no additional improving neighbors can be found.

For this flow and load planning problem, let us first consider neighbors that are defined by reoptimizing the in-tree for each single specific destination terminal (or terminal-delivery day) d. If the in-tree plan selection variables y for all other destinations are fixed at their current values along with the timed dispatch paths for commodities inbound to those locations specified by x, a restricted IP can be solved to search for new values only for variables y^d and x^k for commodities k where $d^k = d$; each such solution is considered a neighbor. Trailer flow variables τ are never fixed in this approach. Note that one idea used to direct to the search toward promising destinations d is to only consider destination terminals for which a large amount of freight is destined; other approaches can be considered that prioritize the reoptimization of larger terminals more frequently.

More specifically, given a current feasible solution $(\bar{x}, \bar{y}, \bar{\tau})$ at some iteration of the search algorithm, $PFLP-TS(d)$ is defined by adding fixing constraints (14.57) and (14.58) to the original formulation (14.48)–(14.56):

$$y_l^u = \bar{y}_l^u \quad \forall u \in N : u \neq d \tag{14.57}$$

$$x_p^k = \bar{x}_p^k \quad \forall k \in K : d^k \neq d \tag{14.58}$$

Erera et al. (2013a) uses this time-expanded network model and solution technique to study flexible flow and load plan designs, including (1) a day-differentiated plan, where an in-tree structure is preserved but the trees are not required to be the same each day of the week, (2) a same-path plan, where the tree requirement is dropped but the freight between two terminals has to follow the same sequence of terminals every day, and (3) an unrestricted plan, where freight is routed without the tree restriction or the same path requirement. Computational experiments demonstrate that high-quality solutions to large-scale problem instances, which represent actual freight volumes transported by the super-regional LTL carrier Saia, can be obtained when limiting the restricted neighborhood search IPs to a solution time limit of 90 s. The study reports cost savings (relative to the initial load plan provided by the carrier) of approximately 4% for traditional load plans and approximately 6.5% for day-differentiated load plans by running the local search for 6 h.

Other IP-based local search neighborhoods may be promising for this problem. One idea is to use an integer program to attempt to attract flow to a specific lane $l \in A$ or to drive flow off of l. Such lane-based neighborhoods may benefit from the fact that they can adjust the in-trees for multiple destinations during a single search iteration; this may be especially important when in-trees for multiple nearby terminals all need to be adjusted simultaneously to remove enough flow from certain time-space dispatch arcs to reduce cost. Another potentially useful idea is to not include the trailer balance constraints (14.53) when solving the neighborhood search integer programs; when the time-space networks are large, there are large numbers of these constraints which can slow the search IP significantly. Instead, we might specify lower bounds on the number of trailers dispatched on some time-space arcs when they have been identified in earlier iterations as useful backhaul lanes for returning empties to outbound-heavy terminals.

Consider an alternative IP-based search approach for solving PFLP-TS where at each iteration, the type of neighborhood (*attract* or *reduce* freight) is chosen as well as a specific geographic lane $l \in A$. Given a lane and a neighborhood type, a set of destination terminals D' is identified whose in-trees may be affected by attracting or removing flow from lane l. For each of these destinations $d \in D'$, a new in-tree is determined via a heuristic as an option to replace the current in-tree to d. The *NewOrOldTree* IP is then solved to search the neighborhood, where for each $d \in D'$ a decision is made to leave its tree unchanged or to adopt the new tree while also choosing new commodity time-space paths compatible with the selections. All in-trees and commodity paths for destinations $d \notin D'$ are fixed and thus remain unchanged.

The success of this approach clearly depends also on the methods used to generate new in-trees. One method useful for the *attract* neighborhood is similar to the IFOL-0 procedure of Powell (1986) described earlier in this chapter. Determining new trees for the *reduce* neighborhood is more complicated, since determining an appropriate new in-tree for d that excludes lane l requires selecting from potentially many feasible choices.

We now describe the *NewOrOldTree* IP, given as follows:

$$\text{minimize} \quad F + \sum_{a \in \mathscr{A}'} c_a \tau_a + \sum_{k \in K'} \sum_{p \in \mathscr{P}'(k)} h_p q_k x_p^k \tag{14.59}$$

subject to

$$\sum_{p \in \mathscr{P}'(k)} x_p^k = 1 \qquad \forall k \in K' \tag{14.60}$$

$$x_p^k \le 1 - z_{d^k} \quad \forall k \in K', \quad \forall p \in \mathscr{P}'(k, OldTree(d^k)) \tag{14.61}$$

$$x_p^k \le z_{d^k} \quad \forall k \in K', \quad \forall p \in \mathscr{P}'(k, NewTree(d^k)) \tag{14.62}$$

$$\sum_{k \in K'} \sum_{p \in \mathscr{P}'(k): a \in p} q_k x_p^k + f_a \le \tau_a \quad \forall a \in \mathscr{A}' \tag{14.63}$$

$$z_{d^k} \in \{0, 1\} \qquad \forall k \in K' \tag{14.64}$$

$$x_p^k \in \{0, 1\} \qquad \forall k \in K', \ \forall p \in \mathscr{P}' \tag{14.65}$$

$$\tau_a \ge MT_a \qquad \forall a \in \mathscr{A}' \tag{14.66}$$

$$\tau_a \in \mathbb{Z}_+ \qquad \forall a \in \mathscr{A}' \tag{14.67}$$

Note that the search formulation (14.59)–(14.67) is no longer simply a restriction of the original integer program. For each destination $d \in D'$, binary decision variable z_d is used to select the new in-tree or old (current) in-tree to d. Time-space paths are (potentially) changed only for some commodities $k \in K'$, where K' denotes all commodities destined for a terminal in D'; note that the current time-space path for some such commodity k remains feasible if the old in-tree is selected, but it may or may not be feasible if the new in-tree is selected for d^k. Let $\mathscr{A}'$ be the subset of time-space dispatch arcs whose trailer flow might change given D' and the specified in-trees. Let F denote all costs associated with trailer movement on arcs $a \in \mathscr{A} \setminus \mathscr{A}'$ and handling for commodities $k \in K \setminus K'$. Constraints (14.61) and (14.62) ensure compatibility of path selection with the new/old tree selections. Constraint (14.63) ensures that enough trailers move along an arc a to carry the freight assigned to the paths passing through a. Here, f_a denotes the sum of the fractional freight for commodities $k \in K \setminus K'$ that will remain moving on dispatch arc a.

Finally, this local search approach uses a different method to model the impact of empty trailer flows on the flow and load plan. Let MT_a be the current minimum number of trailers that must move on arc a in order to guarantee flow balance; this quantity is determined by periodically solving an *empty trailer repositioning minimum cost network flow (MCNF)* formulation given the current x and y solution, and then setting $MT_a = \tau_a$ for any arc a on which empty trailers are planned. The idea here is that the best times and locations to move empty trailers are dictated largely by the underlying freight demand and thus do not need to change frequently, so lower bounds can be used to create useful backhaul opportunities that can be exploited when selecting the flow plan and specific loaded dispatches. Of course, it is also necessary to solve the empty balancing problem at the end of the final iteration to ensure that the final trailer load plan is balanced.

This heuristic is tested in Lindsey et al. (2016) on large-scale problem instances with numerous algorithmic configurations, where a configuration is defined by the rules used in each iteration to (1) decide whether to search an *attract* or a *reduce* neighborhood, and (2) choose the lane l to be used to generate the neighborhood. Computational experiments demonstrate that the approach is effective at generating high-quality solutions in reasonable computation times. Cost reductions from the base flow and load plan of the partner LTL carrier were found in the range of 6–7%.

4.3 Dynamic Discretization Discovery

Solving flow and load planning models that use time-expanded networks to within reasonable provable optimality gaps for the large-scale instances found in practice has been beyond reach for a long time.

However, a novel paradigm, dynamic discretization discovery, has emerged recently as a way to effectively and efficiently find optimal or near-optimal solutions to models using time-expanded networks (Boland et al. 2017). Dynamic discretization discovery allows the solution of such models on a fine discretization without ever fully constructing it. The paradigm has three main components:

- The design of time-indexed IP models based on a *partial discretization of time*, that are efficiently solvable in practice and that yield lower bounds, upper bounds, or exact solutions;
- The design of algorithms that *dynamically discover* partial discretizations, *i.e.,* algorithms that can “refine” a partial discretization of time in order to strengthen the quality of a time-indexed IP model; and
- The design of algorithms that efficiently solve time-indexed IP models.

The latter is stated for completeness sake. In many situations, the use of a standard (commercial or open source) IP solver suffices.

A partially time-expanded network $\mathcal{D}_{\mathcal{T}} = (\mathcal{N}_{\mathcal{T}}, \mathcal{A}_{\mathcal{T}})$ is derived from subsets of the time points that could be modeled at each terminal node. Specifically, we denote the collection of modeled time points as $\mathcal{T} = \{\mathcal{T}_i\}_{i \in N}$, with $\mathcal{T}_i = \{t_1^i, \ldots, t_{n_i}^i\} \subseteq \{1, \ldots, T\}$ representing the time points modeled at terminal node i and T denoting the planning horizon. Given $\mathcal{T}$, the *timed* node set $\mathcal{N}_{\mathcal{T}}$ then has a node (i, t) for each $i \in N$ and $t \in \mathcal{T}_i$.

The timed arc set of a partially time-expanded network consists of arcs of the form $((i, t), (j, \bar{t}))$ where $(i, j) \in A$, $t \in \mathcal{T}_i$, and $\bar{t} \in \mathcal{T}_j$. Note that arc $((i, t), (j, \bar{t}))$ does *not* have to satisfy $\bar{t} = t + t_{ij}$, where t_{ij} is the travel time from terminal node i to terminal node j. In fact, the flexibility to introduce arcs $((i, t), (j, \bar{t}))$ with a travel time that is different from the actual travel time t_{ij} is an essential feature of the partially time-expanded networks, and provides a mechanism to control both the size of the time-expanded network and the approximation properties of the IP model based on it.

Now consider a partially time-expanded version of the generic model presented in Sect. 3.1, *i.e.,*

$$\text{minimize} \quad \sum_{k \in K} \sum_{a \in \mathcal{A}_{\mathcal{T}}} c_a x_a^k + \sum_{a \in \mathcal{A}_{\mathcal{T}}} d_a \tau_a - \sum_{i \in N} \sum_{k \in K \mid o_k = i} h_i q_k \tag{14.68}$$

subject to

$$\sum_{a=((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}} x_a^k - \sum_{a=((j,\bar{t}),(i,t)) \in \mathcal{A}_{\mathcal{T}}} x_a^k = \begin{cases} q_k & \text{if } i = o_k,\ t = e_k \\ -q_k & \text{if } i = d_k,\ t = l_k \\ 0 & \text{otherwise} \end{cases} \forall k \in K, \quad \forall i \in N \tag{14.69}$$

$$x_a = \sum_{k \in K} x_a^k \qquad \forall a \in \mathcal{A}_{\mathcal{T}} \tag{14.70}$$

$$x_a \le Q \tau_a \qquad \forall a \in \mathcal{A}_{\mathcal{T}} \tag{14.71}$$

$$x_a^k \ge 0 \qquad \forall k \in K,\ \forall a \in \mathcal{A}_{\mathcal{T}} \tag{14.72}$$

$$\tau_a \ge 0 \text{ and integer} \qquad \forall a \in \mathcal{A}_{\mathcal{T}} \tag{14.73}$$

Observe that if the time discretization is complete, *i.e.,* $\mathcal{T}_i = \{1, \ldots, T\}$ for $i \in N$, and $\mathcal{A}_{\mathcal{T}}$ consists of all arcs $((i,t),(j,\bar{t}))$ with $\bar{t} = t + t_{ij}$ for $i, j \in N$ and $t = 1, \ldots, T - t_{ij}$, together with all arcs $((i,t),(i,t+1))$ for $i \in N$ and $t = 1, \ldots, T-1$, representing the possibility to wait at terminal node i, then the time-expanded network is acyclic and the above formulation is an exact formulation for the flow planning model.

By choosing $\mathcal{N}_{\mathcal{T}}$ and $\mathcal{A}_{\mathcal{T}}$ carefully, the model may be guaranteed to provide either a lower or an upper bound on the optimal value of flow planning model.

To obtain a lower bound, the concept of a "short" arc is helpful: $((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}$ is *short* if $\bar{t} \le t + t_{ij}$. Three conditions, together, guarantee a lower bound from the IP: (i) $(o_k, e_k) \in \mathcal{N}_{\mathcal{T}}$ and $(d_k, l_k) \in \mathcal{N}_{\mathcal{T}}$ for all $k \in K$, (ii) for all $(i,t) \in \mathcal{N}_{\mathcal{T}}$ and all $j \in N$ with $t + t_{ij} \le l_j$, there exists a $\bar{t}$ with $((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}$, and (iii) every arc in $\mathcal{A}_{\mathcal{T}}$ is short. Note that if the time discretization at terminal node j is quite coarse, it may be that $\bar{t}$ is less than t, suggesting travel backwards in time! Nevertheless, good lower bounds can result. It can be proved that the best such lower bound is obtained by setting $\bar{t} = \max\{t' : t' \le t + t_{ij},\ (j,t') \in \mathcal{N}_{\mathcal{T}}\}$, and permitting no other arc from (i,t) to j to be included in $\mathcal{A}_{\mathcal{T}}$. A partially time-expanded network created in this way has the *longest-arc* property.

A condition that guarantees an upper bound from the IP, provided the IP is feasible, is that all arcs are "long": $((i,t),(j,\bar{t})) \in \mathcal{A}_{\mathcal{T}}$ is *long* if $\bar{t} \ge t + t_{ij}$.

The fundamental idea underlying the dynamic discretization discovery paradigm is to always work with a partial discretization of time so as to ensure that the resulting models can be solved efficiently, but to guarantee that, upon termination of the algorithm, an optimal (continuous-time) solution is (or can be) produced. That is, the idea is to solve a sequence of small IPs, rather than a single large IP.

Thus, whenever the lower-bound IP model does not generate a feasible (and hence optimal) solution, its solution uses some timed arc that is too short. By refining the discretization at the terminals node at the head of that arc, the timed arc that is too short can be made to "disappear", *i.e.,* not be present in the network associated with the refined partial discretization. If the longest-arc property is enforced for the partially time-expanded network constructed at each iteration, the timed arc that is too short, $((i, t), (j, \bar{t}))$ say, can be removed simply by adding $\hat{t}$ to $\mathscr{T}_j$ for any $\hat{t}$ satisfying $\bar{t} < \hat{t} \leq t + \tau_{ij}$. The effect will be to *lengthen* the timed arc, to $((i, t), (j, \hat{t}))$. The natural choice is to take $\hat{t} = t + t_{ij}$.

Excellent computational results for medium-sized instances of the flow and load planning model have been obtained with an interval-based variant of the dynamic discretization discovery algorithm outlined above (Marshall et al. 2020); instances derived from the western operations of a US carrier with about 15 terminals, about 100 load arcs connecting terminals, and about 450 commodities are solved to within 1% of (proven) optimality in about 10 min.

5 Bibliographical Notes

In this section, we provide more detail about specific important papers in consolidation trucking service network design. First, we will discuss some important papers focusing on exact solution approaches useful for flow planning models. Next, we review important papers in the chronology of flow planning for consolidation trucking. We then review papers on flow and load planning and related network design papers that use time-expanded networks. Finally, some discussion of problems downstream from load planning will also be reviewed. These notes are not meant to be a complete and comprehensive chronology, but should provide the reader with a useful initial overview of some of the more important papers in the literature.

The generic arc-based trucking network flow planning model is a multi-commodity capacitated fixed-charge network design (MCND) problem (see e.g., Crainic, 2000). Early work describes Lagrangian approaches for computing lower bounds for cases when the capacity that can be installed on each arc is bounded (see e.g., Crainic et al., 2001). Recently, Chouman et al. (2017) provides an excellent summary of recent exact approaches for this problem class, including those described in Frangioni and Gendron (2009) and Raack et al. (2011). Furthermore, the paper outlines the components of effective cutting plane algorithms for the problem, including one that relies on introducing violated strong inequalities of the form $x_{ij}^k \leq q_k \tau_{ij}$ to improve lower bounds. Evidence is provided that cutting plane algorithms work better with disaggregated commodity representations. Problems solved to optimality or to small gaps are those with at most 100 nodes, 700 arcs, and 400 commodities. Atamtürk and Gunluk (2017) provides a useful review of approaches for the construction of useful classes of valid inequalities for capacitated network design problems, including earlier work in Atamtürk (2002) and Atamtürk and Rajan (2002). Slope-scaling heuristics for these design problems are introduced in Crainic et al. (2004).

An early and important stream of research on flow planning for LTL trucking consolidation networks was initiated in Powell and Sheffi (1983) and Powell (1986); related work using similar flat network models is also covered in Powell and Sheffi (1989) and Powell and Koskosidis (1992). It is important to note here that the work was described as *load planning* (for example, in the title of the 1983 paper), but in the context of these definitions used in this chapter it is best to classify this work as focused on flow planning. Powell (1986) introduces the in-tree flow planning problem for LTL carriers and develops a detailed local search solution heuristic for the problem. Follow-on work in Powell and Koskosidis (1992) constrains plans further by clustering EOL terminals to a primary breakbulk and aligning their flow plans. This paper also presents refined solution approaches to the routing subproblem, including a gradient-based approach for finding primal feasible solutions and subgradient optimization and dual ascent approaches that produce lower bounds and enable estimations of optimality gaps.

The heuristic developed in these papers was implemented originally for flow planning at the U.S. LTL carrier Ryder Truck Lines in an interactive planning system known as APOLLO (Advanced Planner Of LTL Operations). Development continued at Yellow Freight within a system known as SYSNET; Bell et al. (2003) reports that SYSNET was still in use, in an updated version, at Yellow Freight over a decade later. The ideas in these systems were then sold broadly to LTL carriers by the Princeton Transportation Consulting Group (and later Manhattan Associates) within the SuperSPIN system. The software was reportedly used by every major national and regional U.S. LTL carrier in the 1990s and remained in use for nearly 25 years afterwards. Braklow et al. (1992) describes a case study where the software was successfully implemented at Yellow Freight System, and a reduction in number of end-of-line terminals resulted in higher freight density and therefore reduced handling costs and improved service level.

Other authors in this time period developed LTL flow planning models. Roy and Delorme (1989) introduces *NETPLAN*, a nonlinear mixed-integer network optimization model that simultaneously considers flow planning and empty rebalancing using a path-based model. The objective function minimizes the total transportation and consolidation costs, with penalties for overutilization of trailer capacity and failure to meet service standards. An iterative solution methodology (introduced in Crainic and Rousseau 1986) is used to solve the problem. This study, along with Crainic and Roy (1988) and Roy and Crainic (1992), tests the approaches using data from two large Canadian LTL companies and shows that both service offerings could be expanded reliably while also reducing total operating costs.

More recently, Meuffels et al. (2009) addressed a ground transportation consolidation problem for the express package industry using similar network design ideas. This work was conducted for TNT Express, and it is part of a larger group of operations research projects described in Fleuren et al. (2013). In the paper, relatively small tactical express networks are considered for consolidation optimization using flat networks. Additionally, time-feasible schedules for the vehicle fleet are also determined given the consolidation plan.

Time-expanded network models for trucking service network design do not appear in the literature until the 2000s. Prior to this work, some use of models of this type was documented for express package service providers that typically use air and truck movements. Work in this area is described in Barnhart and Schneur (1996), Kim et al. (1999), Barnhart et al. (2002), and Armacost et al. (2002). Decisions in these models include the timed routes of aircraft from (potentially) different fleet types and ground truck transfer decisions to enable service-feasible transfer of packages from origins to destinations. In these papers, integer programming optimization models that use path variables are developed and solved, often relying on column generation for solving large-scale linear programming relaxations.

With increasing demand for faster and time-definite freight transfer due to changing customer service expectations driven in part by the package express industry, LTL carriers now need to plan networks with tighter service guarantees. Jarrah et al. (2009) is the first to consider LTL flow planning with a time-expanded network that allows modeling of explicit service commitments (measured in transit days) to customers that is solved with a slope-scaling approach. The path-based model creates an in-tree flow plan with empty trailer balancing by considering a planning week with a single time-space node on each weekday; in this way, it is not a detailed load dispatch planning model and may still overestimate consolidation opportunities.

Erera et al. (2013a) and Lindsey et al. (2016) model large-scale detailed flow and load planning problems for LTL carriers, following advice from Powell (1986): "Ideally the problem should be formulated as a detailed scheduling problem where the scheduled departure of each tractor would reflect not only a decision that balanced transportation and handling costs but also the actual level of service constraints for each shipment being carried." Both papers introduce integrated flow and load planning integer programming models that use a path-based formulation on a time-space network, and both solve the models using different IP-based local search techniques. Erera et al. (2013a) defines the local search neighborhood by restricting the base integer program to only change flow plan and freight routing variables for a single destination d each iteration, while Lindsey et al. (2016) considers neighborhoods defined by adjusting many in-trees simultaneously (and associated time-space freight paths) to add or remove flow from individual geographic lanes each iteration.

The most modern flow and load planning work in the research literature has focused on developing better approaches for determining the discretization of time-expanded networks. An important new idea is dynamic discretization discovery, where iterative techniques are used to expand the size of a time-expanded network representation of a consolidation trucking network for flow and load planning. A dynamic discretization discovery algorithm for the load plan design problem, which enforces path selection from a set of candidate paths and an in-tree structure, is presented in Hewitt (2019). Follow-on research is developed in Marshall et al. (2020), and unpublished research in this area focuses on adapting principles of dynamic discretization discovery to pragmatic heuristic solution approaches that are often required by large-scale carrier instances.

Other research papers have focused on service network design problems in trucking that lie downstream of flow planning and load planning. One example is the load and dispatch problem considered in Cohn et al. (2007) and Root and Cohn (2008). These papers consider ground package trucking operations for a large carrier given a fixed flow plan. The goal is to build loads and trailer dispatches that meet service requirements, considering both single-trailer and double-trailer combination dispatches using 28-foot pup trailers. Set partitioning models with composite variables that define complete paths for one or more trailers are developed.

Crainic and Roy (1992) focuses on driver scheduling and presents a modeling framework for generating regular driver routes for LTL carriers, given a flow and load plan. The model takes into consideration operational aspects of driver route generation, such as cyclic routes, regular and overtime costs, and maximum permitted duty and working times. The model is solved in three stages: segment generation, route generation, and route selection. Segments are used as the main elements in a set covering model, and a column generation approach is developed. More recently, Erera et al. (2013b) also investigates driver scheduling given a flow and load plan. The paper first introduces the *load plan scheduling problem*, which develops a detailed operational schedule (timed schedule of trailer, tractor, and driver dispatches) required to operate a plan. The approach can be used either with a flow plan only that specifies a geographic transfer sequence of freight for each commodity, or with a flow and load plan where timed trailer dispatches have already been planned. When only a flow plan is given, a detailed load plan is first constructed using a heuristic that sequentially assigns commodities to time-space paths of loads by minimizing path *marginal cost* within a GRASP framework (greedy randomized adaptive search procedure). A key insight in this paper is that some constructed trailer dispatches can be shifted in time in order to improve the driver schedule and its cost, when doing so does not impact the feasibility of the consolidation plan. A novel linear programming formulation is presented to maximize the total width of all trailer dispatch time windows such that the load plan remains feasible. Then, using these adjusted time windows, driver tours are constructed serving each trailer dispatch within its newly-expanded time window using a set covering model and a column generation heuristic.

6 Concluding Remarks and Research Directions

Operations research and service network design models have been used quite effectively for improving motor freight consolidation planning over the past 35 years. Initial successes with frequency-based flat network models for flow planning have now been augmented significantly with flow and load planning models that use detailed time-expanded networks and integrate loaded and empty dispatch planning.

Going forward, there are a number of areas that the field can continue to address to improve service network design for trucking. For example, there is a clear need to better integrate driver and trailer resource planning with flow and load planning. Since driver schedules are highly constrained, it would be best to build plans that recognize that all (or most) dispatches will be covered by driver tours. Crainic et al. (2016) develops initial ideas for effective approaches, and work such as that presented in Hewitt et al. (2019) has developed heuristic approaches that may enable these approaches to be deployed on real-world problems of practical scale.

Another important area for investigation is the set of dynamic planning problems that seek to manage and mitigate uncertainty in both customer demand and supply conditions. Early work in Zhang (2010) focuses on initial ideas for dynamic load planning given updated demand information. Recently, UPS has begun investigation of dynamic load building given primary and alternate freight routing paths for its LTL freight division; fast heuristic approaches are developed in Ridouane et al. (2020) for allocating inbound shipments to scheduled trailer capacity in an effort focused on successfully transferring freight to meet service requirements while also identifying potential scheduled trailerloads for cancellation and cost savings.

Finally, it is also important to extend trucking service network design models to incorporate uncertainty in freight demand. Given that the deterministic planning problems are already very difficult to solve, this is a particular challenge. Early important work in this direction is presented in Lium et al. (2009), and this paper shows using small generic examples that the structure of service designs identified when explicitly modeling uncertainty can be quite different from the designs that result from deterministic models. Baubaid et al. (2018) has more recently considered the stochastic planning problem of setting primary and alternate flow plan paths specifically for LTL freight networks; the paper defines the p-alt planning problem under demand uncertainty, p limits the number of outbound terminals that freight destined to d can flow to next. A 1-alt design represents a standard in-tree flow plan. The approach uses sample average approximation to find p-alt plans that minimize (an approximation of) expected costs (including failure penalties when capacity is not available). Unfortunately, adding multiple scenarios and linking constraints to the already-difficult multi-commodity fixed-charge network design problem severely limits the size of the problems that can currently be addressed by this approach. It may be necessary to develop service network design problems for trucking networks that rely on simpler, approximate problem representations when planning under uncertainty and then to test and refine those designs with more detailed models.

References

Alumur, S., & Kara, B. Y. (2008). Network hub location problems: The state of the art. *European Journal of Operational Research, 190*(1), 1–21.

Armacost, A. P., Barnhart, C., & Ware, K. A. (2002). Composite variable formulations for express shipment service network design. *Transportation Science, 36*(1), 1–20.

Atamtürk, A. (2002). On capacitated network design cut-set polyhedra. *Mathematical Programming, 92*(3), 425–437.

Atamtürk, A., & Gunluk, O. (2017). Multi-commodity multi-facility network design. www.1707.03810.

Atamtürk, A., & Rajan, D. (2002). On splittable and unsplittable flow capacitated network design arc–set polyhedra. *Mathematical Programming, 92*(2), 315–333.

Barnhart, C., & Schneur, R. R. (1996). Air network design for express shipment service. *Operations Research, 44*(6), 852–863.

Barnhart, C., Krishnan, N., Kim, D, & Ware, K. (2002). Network design for express shipment delivery. *Computational Optimization and Applications, 21*(3), 239–262.

Baubaid, A., Boland, N., & Savelsbergh, M. (2018). Dealing with demand uncertainty in service network and load plan design. In W. J. van Hoeve (Ed.), *Integration of constraint programming, artificial intelligence, and operations research. CPAIOR 2018. Lecture notes in computer science* (vol. 10848, pp. 63–71). Berlin: Springer.

Bell, P. C., Anderson, C. K., & Kaiser, S. P. (2003). Strategic operations research and the Edelman prize finalist applications 1989–1998. *Operations Research, 51*(1), 17–31.

Boland, N., Hewitt, M., Marshall, L., & Savelsbergh, M. (2017). The continuous-time service network design problem. *Operations Research, 65*(5), 1303–1321.

Braekers, K., Ramaekers, K., & van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers and Industrial Engineering, 99*, 300–313.

Braklow, J. W., Graham, W. W., Hassler, S. M., Peck, K. E., & Powell, W. B. (1992). Interactive optimization improves service at Yellow Freight System. *Interfaces, 22*(1), 147–172.

Bureau of Transportation Statistics. (2018). Transportation statistics annual report 2018. Tech. Rep., United States Department of Transportation. Washington, D.C. https://doi.org/10.21949/1502596

Campbell, J. F. (1994). Integer programming formulations of discrete hub location problems. *European Journal of Operational Research, 72*(2), 387–405.

Cattaruzza, D., Absi, N., Feillet, D, & González-Feliu, J. (2017). Vehicle routing problems for city logistics. *EURO Journal on Transportation and Logistics, 6*(1), 51–79.

Chouman, M., Crainic, T. G., & Gendron, B. (2017). Commodity representations and cut-set-based inequalities for multicommodity capacitated fixed-charge network design. *Transportation Science, 51*(2), 650–667.

Cohn, A., Root, S., Wang, A., & Mohr, D. (2007). Integration of the load-matching and routing problem with equipment balancing for small package carriers. *Transportation Science, 41*(2), 238–252.

Crainic, T. G., Frangioni, A., & Gendron, B. (2001). Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. *Discrete Applied Mathematics, 112*, 73–99.

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

Crainic, T. G., & Rousseau, J. M. (1986). Multicommodity, multimode freight transportation: A general modeling and algorithmic framework for the service network design problem. *Transportation Research Part B, 20B*(3), 225–242.

Crainic, T. G., & Roy, J. (1988). OR tools for tactical freight transportation planning. *European Journal of Operational Research, 33*(3), 290–297.

Crainic, T. G., & Roy, J. (1992). Design of regular intercity driver routes for the LTL motor carrier industry. *Transportation Science, 26*(4), 280–295.

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

Crainic, T. G., Hewitt, M., Toulouse, M., & Vu, D. M. (2016). Service network design with resource constraints. *Transportation Science, 50*(4), 1380–1393.

Daskin, M. S. (2011). *Network and discrete location: Models, algorithms, and applications*. Hoboken: Wiley.

Drezner, Z., & Hamacher, H. W. (2001). Facility location: Applications and theory. Berlin: Springer.

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

Erera, A. L., Hewitt, M., Savelsbergh, M. W., & Zhang, Y. (2013b). Creating schedules and computing operating costs for LTL load plans. *Computers and Operations Research, 40*(3), 691–702.

Farahani, R. Z., Hekmatfar, M., Arabani, A. B., & Nikbakhsh, E. (2013). Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering, 64*(4), 1096–1109.

Fleuren, H., Goossens, C., Hendriks, M., Lombard, M., Meuffels, I., & Poppelaars, J. (2013). Supply chain-wide optimization at TNT express. *Interfaces, 43*(1), 5–20.

Frangioni, A., & Gendron, B. (2009). 0–1 reformulations of the multicommodity capacitated network design problem. *Discrete Applied Mathematics, 157*(6), 1229–1241.

Golden, B. L., Raghavan, S., & Wasil, E. A. (2008). *The vehicle routing problem: Latest advances and new challenges* (vol. 43). Berlin: Springer.

Hewitt, M. (2019). Enhanced dynamic discretization discovery for the continuous time load plan design problem. *Transportation Science, 53*(6), 1731–1750.

Hewitt, M., Crainic, T. G., Nowak, M., & Rei, W. (2019). Scheduled service network design with resource acquisition and management under uncertainty. *Transportation Research Part B: Methodological, 128*, 324–343.

Jarrah, A. I., Johnson, E., & Neubert, L. C. (2009). Large-scale, less-than-truckload service network design. *Operations Research, 57*(3), 609–625.

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

Lindsey, K., Erera, A., & Savelsbergh, M. (2016). Improved integer programming-based neighborhood search for less-than-truckload load plan design. *Transportation Science, 50*(4), 1360–1379.

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

Love, R. F., Morris, J. G., & Wesolowsky, G. O. (1988). *Facilities location*. New York: Elsevier.

Marshall, L., Boland,. N., Savelsbergh, M., & Hewitt, M. (2021). Interval-based dynamic discretization discovery for solving the continuous-time service network design problem. *Transportation Science, 55(1), 29–51.*

Meuffels, I., Fleuren, H., Cruijssen, F., & Dam, E. (2009). Enriching the tactical network design of express service carriers with fleet scheduling characteristics. *Flexible Services and Manufacturing Journal, 22*, 3–35.

Mirchandani, P., & Francis, R. (1990). *Discrete location theory*. *Wiley-Interscience series in discrete mathematics and optimization*. Hoboken: Wiley.

O'Kelly, M. E. (1986). The location of interacting hub facilities. *Transportation Science, 20*, 92–106.

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

Powell, W. B., & Koskosidis, I. A. (1992). Shipment routing algorithms with tree constraints. *Transportation Science, 26*(3), 230–245.

Powell, W. B., & Sheffi, Y. (1983). The load planning problem of motor carriers: Problem description and a proposed solution approach. *Transportation Research Part A: General, 17*(6), 471–480.

Powell, W. B., & Sheffi, Y. (1989). Design and implementation of an interactive optimization system for network design in the motor carrier industry. *Operations Research, 37*(1), 12–29.

Psaraftis, H. N., Wen, M., & Kontovas, C. A. (2016). Dynamic vehicle routing problems: Three decades and counting. *Networks, 67*(1), 3–31.

Raack, C., Koster, A. M., Orlowski, S., & Wessäly, R. (2011). On cut-based inequalities for capacitated network design polyhedra. *Networks, 57*(2), 141–156.

Ridouane, Y., Herszterg, I., Boland, N., Erera, A., & Savelsbergh, M. (2020). Near real-time loadplan adjustments for less-than-truckload carriers. Optimization Online. http://www.optimization-online.org/DB_HTML/2020/01/7565.html

Root, S., & Cohn, A. (2008). A novel modeling approach for express package carrier planning. *Naval Research Logistics, 55*, 670–683.

Roy, J., & Crainic, T. G. (1992). Improving intercity freight routing with a tactical planning model. *Interfaces, 22*(3), 31–44.

Roy, J., & Delorme, L. (1989). NETPLAN: A network optimization model for tactical planning in the less-than-truckload motor-carrier industry. *INFOR Journal, 27*(1), 22–35.

Savelsbergh, M., & van Woensel, T. (2016). 50th anniversary invited article–city logistics: Challenges and opportunities. *Transportation Science, 50*(2), 579–590.

Snyder, L. V. (2006). Facility location under uncertainty: A review. *IIE Transactions, 38*(7), 547–564.

Zhang, Y. (2010). Advances in LTL load plan design. Ph.D. Thesis, Georgia Institute of Technology.