Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

An Integrated Approach to Tactical Transportation Planning in Logistics Networks

Tobias Harks, Felix G. König, Jannik Matuschke, Alexander T. Richter, Jens Schulz

TRANSPORTATION SCIENCE
Articles in Advance, pp. 1–22
ISSN 0041-1655 (print) | ISSN 1526-5447 (online)

http://dx.doi.org/10.1287/trsc.2014.0541

An Integrated Approach to Tactical Transportation Planning in Logistics Networks

Tobias Harks
School of Business and Economics, Maastricht University, Maastricht, The Netherlands,
t.harks@maastrichtuniversity.nl

Felix G. König
TomTom International BV, 12435 Berlin, Germany, felix.koenig@tomtom.com

Jannik Matuschke, Alexander T. Richter, Jens Schulz
Technische Universität Berlin, Institut für Mathematik, 10623 Berlin, Germany
{matuschke@math.tu-berlin.de, arichter@math.tu-berlin.de, jschulz@math.tu-berlin.de}

We propose a new mathematical model for transport optimization in logistics networks on the tactical level. The main features include accurately modeled tariff structures and the integration of spatial and temporal consolidation effects via a cyclic pattern expansion. Using several graph-based gadgets, we are able to formulate our problem as a capacitated network design problem. To solve the model, we propose a local search procedure that reroutes flow of multiple commodities at once. Initial solutions are generated by various heuristics, relying on shortest path augmentations and LP techniques. As an important subproblem we identify the optimization of tariff selection on individual links, which we prove to be *NP*-hard and for which we derive exact as well as fast greedy approaches. We complement our heuristics by lower bounds from an aggregated mixed-integer programming formulation with strengthened inequalities. In a case study from the automotive, chemical, and retail industries, we prove that most of our solutions are within a single-digit percentage of the optimum.

Keywords: logistics; freight transportation; modeling; capacitated network design; local search; mixed-integer programming
History: Received: September 2012; revision received: June 2013; accepted: November 2013. Published online in *Articles in Advance*.

1. Introduction

The ongoing globalization of markets over the past decades accounts for an ever-increasing shipping volume of goods worldwide. In all industries, companies operate facilities spread across the world to maximize profitability, and procurement and distribution have become global operations. The ensuing demand for transportation has fostered the growth of huge international logistics networks with the potential to increase efficiency through economies of scale, pooling of orders, and a global view on network layout.

The task of designing and operating such logistics networks belongs to the broad realm of *supply chain management* (*SCM*), "the management of flows between and among all stages of a supply chain to maximize total profitability" (Chopra and Meindl 2007, p. 6). As this very general definition indicates, SCM addresses a multitude of issues ranging from location, product, and marketing decisions to the management of information exchange and coordination across different stages of the supply chain. *Transportation planning* in particular occupies a central place in SCM, as transport and storage of physical goods account for a significant share of the operational cost in a supply network.

Because of the strong variance in lead times associated with the different decisions to be made in SCM, the planning process is naturally structured hierarchically in strategic, tactical, and operational levels (Simchi-Levi, Kaminsky, and Simchi-Levi 2003). The work presented in this paper is concerned with transportation planning on the *tactical level*. Here, it is commonly assumed that the supply chain is already in place: location and product decisions have been made, and the general design of the supply chain network is fixed. Typical logistics decisions on the tactical level include the amount of flow between the existing nodes of the network, e.g., which customers to serve from which warehouses or suppliers, how much inventory to keep at which locations, and which transportation modes and delivery frequencies to employ on the different connections (Geunes and Pardalos 2003).

This paper proposes an approach to model and solve the key tasks in tactical transportation planning in an integrated fashion, explicitly including realistic transport tariffs and the tradeoff between inventory cost and economies of scale in transportation.

1.1. Problem Description

We give a general description of the task we refer to as tactical transportation planning and introduce some terminology we will use throughout this paper. We consider a network of *facilities*, which are of different types, like production plants, warehouses, distribution centers, or retailers. Some facilities have a *supply* of or a *demand* for certain products, also known as *commodities*, which can be numerous and very different, e.g., in their mass, volume, or value. Facilities are joined by *transport relations*, and on each transport relation, different *transport tariffs* are available corresponding to concurring offers of freight forwarders and available transportation modes. Each transport tariff is characterized by capacity restrictions and a cost function, describing how much of a commodity (or of some commodity mix) can be transported, and at which cost. For example, a full truck load (FTL) tariff may have a certain truck type's payload and footprint as capacity restrictions and incur a fixed charge cost. Some facilities may be able to carry *inventory*, usually with a commodity-dependent capacity and cost. *Handling cost* may result from commodities passing through a facility, such as a distribution center, regardless of whether they are moved to inventory or not.

Quite commonly, transportation cost includes fixed-charge costs for dispatching shipments, and the larger a shipment, the lower the effective per unit shipping cost. Hence, a key ingredient to successful tactical planning in a logistics network is the efficient *consolidation* of material flows, i.e., the combination of smaller order amounts into larger shipments to utilize capacity efficiently and enable economies of scale (Çetinkaya 2005). Consolidation may occur over space as well as time. In *spatial* consolidation, material flows of different origins are accumulated at one node and forwarded jointly to the next. In *temporal* consolidation, material is kept in inventory at a node for some time until more flow arrives, thereby enabling a larger outbound shipment. Because holding inventory also incurs cost, there is a tradeoff to be considered here.

This interplay between inventory cost and different transport tariffs necessitates a notion of time in planning. Since temporal details such as transport transit times or demand deadlines are commonly postponed to operational planning, the goal in tactical optimization is a cyclic *pattern* of deliveries and inventory. The length and structure of this pattern usually follows some natural notion of rough timing, such as "once every month," "once every week," or "once every day of the week," and in each slot of the pattern (e.g., each month, week, or weekday), deliveries are dispatched and inventories replenished or depleted.

All in all, the outcome of tactical transportation planning as described here comprises

- The paths each commodity takes through the network from its sources to its sinks, i.e., the total amount of flow for each commodity on each transport relation
- The transport tariffs employed on each transport relation, together with an assignment of a commodity mix to each of them
- A cyclic pattern in which transports are executed for each tariff used on each transport relation, including the amounts shipped for each commodity in each slot of the pattern
- A pattern of inventory levels for each commodity at each node, supporting the above transport patterns.

Again, note that in tactical planning, the aim is not to use the results to operate the logistics network directly, as this is the subject of operational planning. Rather, tactical optimization intends to aid with decisions that have to be made with some lead time, providing the framework for efficient operation: How much throughput capacity needs to be reserved at certain distribution centers? Which logistics provider should be cooperated with on which network connections, and which available tariffs will be employed on what volume of commodities? Hence, the main purpose of many details in tactical modeling is not primarily to reflect operational reality, but much more to yield a realistic assessment of operational cost in the framework provided.

1.2. Our Contribution and Overview of the Paper

In §2, we propose a new model for the optimization of transportation networks on the tactical level. In our model, different commodities are flexibly characterized in terms of their properties (like mass, volume, and value), and a choice of many different transportation modes and tariffs is naturally incorporated, with capacities and costs accurately reflecting the properties of the (mix of) commodities transported. Moreover, our model includes the possibility for flexible, cyclic delivery patterns on each network connection, accurately modeling the tradeoff between inventory cost and economies of scale in transportation. Although we assume that location decisions have been made and facilities are already in place, the main decision variables of our model include the flow paths of commodities through the network, the transportation tariffs, and inventory levels. Using several graph-based gadgets, we are able to formulate our problem as a network design problem. Note that in contrast to the broad literature on classical network design problems (see §1.3 for concrete pointers to the literature), our formulation integrates different realistic transportation tariffs, cyclic delivery patterns, and inventory costs all in one model.

The resulting network design problem can be naturally formulated as a mixed-integer programming

(MIP) model, but the precise replication of complex tariff structures (via the previously mentioned gadgets) leads to a drastically increased number of variables, putting basic MIP approaches out of reach (at least for instances arising in practice). We identify the problem of selecting optimal tariffs on a single transport relation as an important subproblem that is crucial in speeding up the solution process: to identify cost-efficient paths, our algorithms need good and fast estimates on the cost incurred in sending a particular amount of flow along a transport relation. These cost estimates are performed very frequently (easily more than a million times during the optimization of a single network) and therefore need to be carried out even faster. In §3, we propose different algorithms that provide an efficient balance of accuracy and speed for solving this *NP*-hard subproblem.

In §4, we propose a local search heuristic that employs local changes on a path decomposition of flow in the network using the previously mentioned tariff selection subroutines. In contrast to many local search heuristics known in the literature (that either work directly on the design variables or reroute the flow of a single commodity only), our approach applies a neighborhood search based on path decomposition of flow and rerouting multiple commodities simultaneously. To obtain good initial solutions for our local search heuristic, we provide two successive shortest path type algorithms. The first method is designed with an emphasis on speed and low memory requirement and is able to generate solutions of reasonable quality for even the largest instances in a short time. The second is more accurate in cost estimation and therefore is used as the central subroutine in our local search improving moves. By forbidding certain paths (e.g., direct connections) and linearizing costs we further tune the initial solutions toward a high level of flow consolidation that will eventually be disaggregated by the local search heuristic.

In §5, we complement our heuristic approach using MIP techniques. As the plain MIP formulation introduced earlier is not suited for solving reasonably sized real-world instances because of enormous problem sizes, we propose an aggregated formulation that considerably reduces model size and still yields good dual bounds. We combine this with efficient preprocessing techniques to tighten the relaxation and a postprocessing step to improve solution quality. Combining the linear programming (LP) relaxation of this strengthened and aggregated formulation with the tariff selection heuristics mentioned earlier yields a third way of constructing initial solutions for our local search procedure, which shows best final results on average.

In §6, we evaluate the performance of our different algorithmic approaches on a library of real-world instances provided by our project partner 4flow AG, a logistics consultancy company. The test set consists of case studies from the automotive, chemical, and retail industries with up to thousands of locations and hundreds of commodities. We can prove that most of our solutions are within a single-digit percentage of the optimum and that our modeling and algorithmic techniques yield a cost reduction of more than 10% over the current status quo, which could result in an annual savings of several million euros.

1.3. Related Work

Mathematical optimization for logistics problems has been a vast field of research for several decades. We give an overview over models and algorithms for tactical transportation planning.

1.3.1. Models for Transportation Planning. An excellent overview of network-based optimization techniques for SCM is given by Geunes and Pardalos (2003). The authors review articles dealing with strategic as well as tactical and operational planning.

In one of the earliest optimization models for SCM, Geoffrion and Graves (1974) model a multicommodity network with several plants, possible distribution center locations, and demand zones on the strategic level. Although the model incorporates fixed location costs, as well as upper and lower bounds on the throughput of a distribution center, it does not consider inventory decisions and assumes transportation costs to be linear. The resulting MIP model is solved using Benders decomposition. A strategic optimization model that incorporates the interdependence of location, transportation, and inventory decisions is described by Jayaraman (1998). Here, different transportation modes can be chosen for each connection in the network. Each mode is associated with a commodity-dependent per unit cost and a *delivery frequency*. Keeping inventory at a plant or warehouse incurs per unit inventory cost, and the amount of inventory held results from the delivery frequencies of the outbound transportation modes used. Note that this still captures temporal consolidation rather coarsely, as theoretically, transportation modes with low delivery frequency could also carry low shipping volume, making their assumed low per-unit cost unrealistic. The model is solved using standard MIP solvers.

The above networkwide SCM models are focused on strategic planning and incorporate location decisions, but the tactical and operational tradeoff between transportation and inventory cost lies at the heart of *dynamic lot-sizing* in inventory theory. In the basic version of dynamic lot-sizing introduced by Wagner and Whitin (1958), different demands for a commodity at one facility need to be met in multiple periods. In each period, an arbitrary amount can be ordered at fixed per order cost, whereas per unit inventory cost is incurred. The goal is to determine the amount ordered in each period such that all demands are met on time and the sum of order and inventory costs are minimized. This basic model has been extended in many ways since that study, and most variants are computationally hard; see, e.g., Jans and Degraeve (2007) for an overview. The practical importance of considering the tradeoff between transportation and inventory cost is highlighted impressively by Burns et al. (1985) and Blumenfeld et al. (1987); these authors were able to reduce logistics costs by 26% in a case study for General Motors.

Generalizing lot-sizing to networks with multiple stages brings it closer to the requirements of tactical transportation planning. The first such model was introduced by Clark and Scarf (1960) and further developed by Afentakis, Gavish, and Karmarkar (1984) and Afentakis and Gavish (1986). An overview of more recent works can be found in Stadtler (2003). Most of these models, however, still make rather restrictive assumptions on the structure of the network considered and transportation costs incurred. Moreover, the quantity of material flowing between node pairs is fixed a priori in all lot-sizing models, so the possibility for more spatial consolidation at hubs is effectively ignored.

Kempkes, Koberstein, and Suhl (2010) propose a general model for the integrated operational planning of external and internal logistics of the last two stages of a supply chain. In their model, all costs depend on the usage of resources, like mass or volume; this dependence can be piecewise constant as well as linear and may involve multiple resources. Planning occurs over multiple noncyclic periods, and inventory cost is taken into account. The authors devise a flow-based construction heuristic to generate an initial feasible solution that is passed to a standard MIP solver. To introduce all details necessary for realistic operational planning, their model even allows for logical relations between different resources; however, this significantly increases the algorithmical challenge of solving large-scale instances. Accordingly, their solution approaches are validated on relatively small instances involving only five planning periods with networks of up to 25 nodes, several hundred edges, and up to 100 commodities.

In a more tactical context, Schöneberg, Koberstein, and Suhl (2010) propose a similar resource-based model for optimizing the choice of delivery profiles in *area forwarding-based networks*. In such networks, suppliers are grouped into areas and each area is equipped with a consolidation center run by a logistics carrier. The main decision variables are the choices from a fixed set of delivery profiles for each supplier and the use of vehicles on the main legs (i.e., the connections between consolidation centers and the target). The authors propose a solution method that first decomposes the model by fixing certain decisions for each possible delivery profile and then generates an initial feasible solution for the MIP solver using a two-phase construction heuristic. The approach is evaluated in the logistics network of a German truck manufacturer, achieving cost savings of up to 36% in individual areas.

The model introduced in this paper, as well as the models by Kempkes, Koberstein, and Suhl (2010) and Schöneberg, Koberstein, and Suhl (2010), are based on capacitated network design formulations (see §1.3.2). An alternative approach to modeling nonlinear transportation tariffs are concave-cost network flows; see Guisewite and Pardalos (1990) for a survey. Note also that all three models mentioned above include the possibility of concave cost functions (see §2.4.2 how they can be modeled in the context of the present work).

In contrast to the model of Kempkes, Koberstein, and Suhl (2010), our approach focuses exclusively on transportation planning. Thus, it does not consider globally interdependent resources, making it possible to encapsulate tariff selection in a local subproblem and allowing for larger transportation networks to be solved. It also differs from the model by Schöneberg, Koberstein, and Suhl (2010), which employs delivery profiles to model replenishment cycles, whereas the present paper is concerned with *dynamic planning* and also allows for more general networks with multiple levels of intermediate hubs instead of two-layered area forwarding-based networks.

1.3.2. Capacitated Network Design. Although network flow seems to be the dominant aspect in logistics network optimization, the fixed-cost nature of transport tariffs brings in network design decisions: we have to install sufficient capacity in the network such that all flow can be routed. In the literature, such mixtures of network flow and network design are referred to as *capacitated network design* or *fixed-charge network flow* and are widely used for models not only in logistics but also in telecommunications and infrastructure planning (see the surveys by Magnanti and Wong 1984 and Crainic 2000).

Most capacitated network design problems are very hard to solve both in theory and in practice. In fact, the model presented in this paper generalizes several problems that are not only *NP*-hard but even highly inapproximable from a theoretical point of view, e.g., the single-pair version of the capacitated survivable network design problem, for which Chakrabarty et al. (2011) showed that it does not even permit an approximation factor of $2^{\log^{1-\varepsilon}(n)}$ for any $\varepsilon > 0$ (unless all problems in *NP* can be solved in quasipolynomial time). Furthermore, *NP*-hardness still holds for very basic and sparse classes of networks like so-called series-parallel graphs because a version of the multiple

Steiner subgraph problem (Richey and Parker 1986) can be reduced to our model.

This intrinsic hardness, combined with the enormous size of instances encountered in practical applications from logistics contexts, leaves little hope for exact solution approaches that run in acceptable time. Therefore, fast combinatorial heuristics appear to be the method of choice. The current state of the art is mainly built on specialized tabu search procedures. Crainic, Gendreau, and Farvolden (2000) proposed a tabu search procedure based on a neighborhood in the multicommodity flow polytope. Their algorithm was later adapted for parallelization by Crainic and Gendreau (2002). A different neighborhood for tabu search was introduced by Ghamlouche, Crainic, and Gendreau (2003), operating on the network design and modifying it along cycles. The same authors refine this procedure by supplementing it with a path-relinking technique (Ghamlouche, Crainic, and Gendreau 2004).

A different approach for solving fixed-charge network flow problems is constituted by *slope scaling*. The slope scaling procedure, first proposed by Kim and Pardalos (1999) for single-commodity fixed-charge network flow, iteratively solves the min-cost flow problem arising from linearizing the fixed costs according to the current solution. Crainic, Gendron, and Hernu (2004) generalize this technique to multicommodity capacitated network design and augment it by Lagrangian perturbation and intensification/diversification mechanisms based on a long-term memory.

All algorithms referenced above are designed for general capacitated network design problems and have been successfully tested on a standard benchmark set of randomly generated instances of moderate size with at most 100 nodes and 400 edges, introduced by Crainic, Gendreau, and Farvolden (2000).

1.3.3. MIP Approaches to Network Design. Several exact solution techniques for capacitated network design have been studied; see, e.g., the survey by Costa (2005). These techniques range from Lagrangean relaxation over column generation to Benders decomposition.

Kliewer and Timajev (2005) integrate cover inequalities and local cuts in a Lagrangean-based lower bound, whereas Frangioni and Gendron (2009) study a 0-1-reformulation for piecewise linear costs and show the computational benefits of strong linking inequalities. Chouman, Crainic, and Gendron (2011) present lifting procedures for *strong capacity* and *network cutset inequalities* for fixed-charge network flow problems. Another promising technique to solve capacitated network design problems is to apply a Benders decomposition; see, e.g., Costa (2005), Cakir (2009). Costa, Cordeau, and Gendron (2009) show the relation between different classes of inequalities. In particular, the authors explain how the inequalities from (nonextreme) dual rays of the Benders framework and cutset inequalities can be strengthened via shortest path computations to become metric inequalities. To improve the running times, Fischetti, Salvagnin, and Zanette (2010) propose to find a minimal infeasible subsystem. They show that this idea can be integrated into the subproblem heuristically.

These works indicate that the scope of tractable instance sizes for these methods is roughly limited to 30 nodes, 500 edges, and 200 commodities; i.e., for the few larger instances reported on, the provable gaps on solution quality exceed single digits.

2. Mathematical Model

Our model, which we call TTP for tactical transportation planning, is at its heart based on multicommodity network flow, with both linear and fixed-charge costs on the edges. However, we extend the standard concepts of capacity and cost to more generality in order to reflect the requirements of logistics modeling more precisely. Moreover, we expand the underlying network significantly to model delivery patterns, inventory effects, and complex transport tariffs. We detail all of these features in the following sections.

2.1. Pattern Expansion

The tradeoff between minimizing inventory cost and taking advantage of the economies of scale in transportation is of key importance in tactical logistics planning. Temporal and spatial consolidation effects regularly determine which tariff is most suitable on a connection. Consequently, even the decision of which path in the network is most efficient for a commodity may ultimately depend on temporal delivery patterns. As tactical planning defines the environment for operational planning that will take place repeatedly over time, a solution should be a *cyclic* pattern for dispatching deliveries and replenishing and depleting inventories. To integrate temporal and spatial consolidation with cyclic delivery patterns, we introduce the notion of *pattern-expanded networks*.

A pattern-expanded network denoted by $\mathcal{G}$ has two main components. The first is the *base network* B, which comprises the physical entities of the transport network: facilities (or *nodes*) together with corresponding *transport relations* between facilities. The second parameter is a cycle length F defining the number of time slots (e.g., 7, 30, or 356 days) available in a period. The pattern-expanded network $\mathcal{G}$ is now obtained from B and F by introducing F copies of B denoted by $B_1, \ldots, B_F$ and connecting copies of each node of every two adjacent networks B_i and B_{i+1} by directed *holdover edges* (the direction is from nodes in B_i to those in B_{i+1}). Moreover, the nodes of the last copy B_F are also connected by holdover edges to their corresponding copies in the first copy B_1, thus giving a cyclic network structure. If commodities are sent along holdover edges from B_F to B_1, this corresponds to storing commodities at the corresponding nodes at the end of a cycle to the beginning of the next cycle. Costs can be associated with holdover edges modeling inventory costs. In the following, we will not differentiate between holdover edges and transport edges. We denote the set of nodes in the pattern-expanded network by $\mathscr{V}$ and the set of all edges (also called transport relations) of $\mathscr{G}$ by $\mathscr{R}$.

We illustrate this cyclic construction with an example. Consider the base network in Figure 1(a) involving two source-sink pairs (s_1, t_1) and (s_2, t_2). In this example, we chose $F = 3$; i.e., transports may occur only in three time slots, e.g., three days a week. The pattern-expanded network now involves the three copies of the base network and the additional holdover edges, as illustrated in Figure 1(b).

2.2. Commodities and Properties

Commodities in a logistics network can be very diverse—e.g., in their size, weight, or value—and logistics costs and transport capacities cannot be realistically assumed to be oblivious to this diversity and the resulting interdependencies when mixing commodities in transport. We introduce the concept of flexible *properties* to characterize commodities. A set of commodities K and a list of relevant properties P are parameters of our model. Each commodity $i \in K$ is assigned a per unit extent α_{ij} for each property $j \in P$. The main motivation for introducing these properties is that transportation costs (introduced in §2.3) will mostly depend on the total extent of each property of a commodity mix (rather than the specific type of commodities itself), thus reflecting the effects of consolidating goods for utilizing vehicle capacities more efficiently.

In the following, a mix of commodities will be denoted by a *commodity vector* $x \in \mathbb{R}_+^K$, and the *aggregated properties* of such a mix x are expressed by $\alpha(x) \in \mathbb{R}_+^P$ with $\alpha_j(x) := \sum_{i \in K} \alpha_{ij} x_i$.

Each node in the pattern expanded network may have a supply of, or a demand for, certain commodities. These supplies and demands are expressed by a balance vector $b(v) \in \mathbb{R}^K$ for each node $v \in \mathscr{V}$ (note that these values might be different even for distinct copies of the same node in the base network). A node with a supply $(b_i(v) > 0)$ of a certain commodity $i \in K$ is called a *source* of i; a node with a demand $(b_i(v) < 0)$ is called a *sink* of i. The goal is to transport all supplies from the sources to the sinks, satisfying all demands.

2.3. Transport Tariffs

When shipping goods on a transport relation, different *transport tariffs* are available. For each transport relation $R \in \mathscr{R}$ we denote by $T(R)$ the set of available tariffs for transporting a flow of commodities from start(R) to end(R). Each such tariff $t \in T(R)$ is associated with a cost function $C_t: \mathbb{R}_+^K \to \mathbb{R}_+$. We also assume that all cost functions fulfill the economies of scale principle, i.e.,

$$C_t(a+b) \le C_t(a) + C_t(b). \tag{1}$$

A solution of our model consists of a multicommodity flow in the pattern-expanded network satisfying all demands, together with an assignment of the flow on each transport relation to the tariffs available on this relation. More formally, let $x(R) \in \mathbb{R}_+^K$ denote the total (multicommodity) flow to be shipped on transport relation $R \in \mathscr{R}$, and let $x(t) \in \mathbb{R}_+^K$ denote the amount of flow transported using tariff $t \in T(R)$. Then our goal is to find an optimal solution to

$$\min \sum_{R \in \mathscr{R}} \sum_{t \in T(R)} C_t(x(t))$$

$$\text{s.t.} \sum_{R \in \delta^+(v)} x_i(R) - \sum_{R \in \delta^-(v)} x_i(R) = b_i(v) \quad \forall v \in \mathscr{V}, \forall i \in K,$$

(a) Base network   (b) Pattern-expanded network

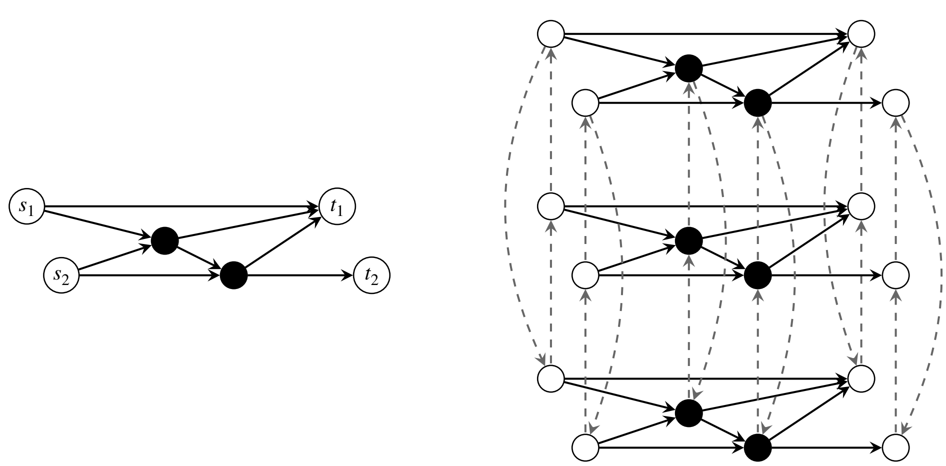


Figure 1 Network Expansion

Notes. Base network with associated pattern expanded network. Dashed edges denote holdover edges.

$$\sum_{t\in T(R)} x_i(t) = x_i(R) \quad \forall\, R\in\mathcal{R},\ \forall\, i\in K,$$

$$x(t)\geq 0 \quad \forall\, t\in T(R),\ \forall\, R\in\mathcal{R},$$

where $\delta^+(v)$ and $\delta^-(v)$ denote the sets of outgoing and incoming arcs of node v, respectively.

We will now present a set of cost functions that covers most tariffs occurring in today's logistical applications. In §2.4, we will show how all of these cost functions can also be modeled in a unified form as a capacitated network design problem.

Linear costs. In many logistical applications, commodity-dependent linear costs of the form

$$C(x) = \sum_{i\in K} c_i \cdot x_i$$

with cost rates $c_i \in \mathbb{R}_+$ for each commodity occur, e.g., in the form of handling costs, in-stock and in-transit inventory costs and simple linear tariffs without interdependencies of the transported commodities.

Maximum over multiple cost rates. Tariffs can also be specified as the maximum over varying cost rates for distinct properties, i.e., when sending a shipment that rate applies for which the cost is highest. More formally, with c_j being the cost rate for property j, the cost function is given as

$$C(x) = \max_{j\in P}\left\{c_j \cdot \sum_{i\in K}\alpha_{ij}x_i\right\}.$$

Note that, in contrast to the linear costs described in the preceding paragraph, these maximum cost functions capture the effect of cost savings when mixing commodities of different dimensions, e.g., light but voluminous with heavy but compact ones.

Property-dependent piecewise constant costs. Many tariffs, such as those offered by most FTL carriers and some less than truck load (LTL) carriers, are based on piecewise constant cost functions; i.e., they are specified by a cost $c\in\mathbb{R}_+$ and a capacity vector $\beta\in\mathbb{R}_+^P$ for a single shipment, yielding the function

$$C(x) = c\cdot \max_{j\in P}\lceil \alpha_j(x)/\beta_j\rceil.$$

In practice, logistics carriers offer groups of such tariffs realizing different levels of discounts for higher shipment volumes. We will see in §3 that finding the most cost-efficient combination of such tariffs for a given shipment volume is already an *NP*-hard problem.

Of course, linear and fixed costs can also occur at the same time, e.g., to model a transport to a distribution center that incurs a fixed cost for transportation and a linear cost for handling the incoming shipment at the distribution center. We thus also allow the combination of these two cost types.

Incremental discount costs. We consider a tariff with varying cost rates, depending on a single property. The cost rates are specified on intervals and decrease with increasing size of shipment, resulting in a piecewise linear and concave cost function; see Table 1 for an illustration. Formally, we label the intervals from 0 to L. For each $l\in[L]$, let $c^{(l)}\in\mathbb{R}_+$ be the cost rate on the interval $[\beta_j^{(l)}, \beta_j^{(l+1)})$ for the fixed property $j\in P$, with $0=\beta_j^{(0)}<\beta_j^{(1)}<\cdots<\beta_j^{(L)}<\beta_j^{(L+1)}=\infty$ and $c^{(0)}>c^{(1)}>\cdots>c^{(L-1)}>c^{(L)}$. Then the cost function is

$$C(x) = \sum_{l=0}^{L} c^{(l)}\cdot \min\{\beta_j^{(l+1)}-\beta_j^{(l)}, (\alpha_j(x)-\beta_j^{(l)})^+\}.$$

All-unit discount costs. Again we consider linear cost rates in some property $j\in P$ with several levels of decreasing per unit cost rates. Different from the above, however, a cost rate applies to the entire transport volume as long as it lies within the corresponding interval. To ensure monotonicity, a cost cap applies whenever the cost with respect to the current rate exceeds the cost at the beginning of the next level—this corresponds to the common practice of declaring higher volumes than are actually transported in such cases (Chan et al. 2002). See Table 1 for an illustration of the resulting cost function. Formally, if cost rate $c^{(l)}$ for $l\in[L]$ is applicable starting from transport volume $\beta_j^{(l)}$ on, the cost function is

$$C(x) = \min_{l\in[L]}\big(c^{(l)}\cdot \max\{\alpha_j(\tilde{x}), \beta_j^{(l)}\}\big).$$

2.4. Reformulation as Capacitated Network Design

We will now provide a different perspective to the model presented in the previous section. We introduce the concept of *containers* to model the different types of tariffs in a way that leads to a unifying description of the above model as a fixed-charge multicommodity flow problem. A natural formulation as a mixed-integer program can easily be obtained from this description, making it accessible to MIP-based solving techniques, whereas its compact structure effectively demonstrates the degree of mathematical uniformity achieved in modeling.

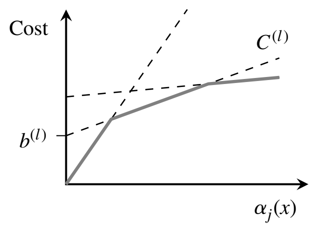
We first present the alternative formulation of the model to its full extent and then show the equivalence to the formulation in the previous section by describing how different cost functions can be modeled using containers.

2.4.1. The Tariff-Expanded Network. For each tariff on a transport relation, we introduce a *gadget* consisting of edges, which connects the start node of the relation with its end node. On each edge, a certain type of *container* is available, and capacities can be installed on the edge in increments of this container type. After replacing all transport relations in

Table 1 Modeling Complex Transport Tariffs with Containers

Tariff | Cost | Gadget

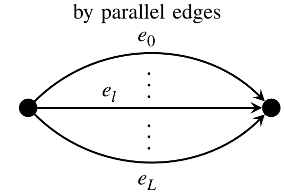
Incremental discount (piecewise linear concave)



$$C(x) = \min_{l\in[L]} C^{(l)}(x)$$

$$C^{(l)}(x) := c^{(l)}\alpha_j(x) + b^{(l)}$$

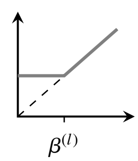
Minimum modeled by parallel edges



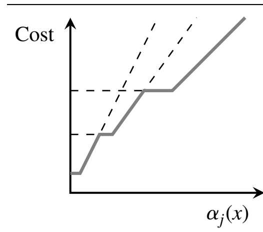
$$c(e_l) = b^{(l)}$$

$$c_i(e_l) = \alpha_{ij} c^{(l)}$$

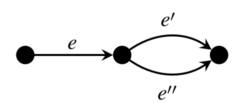
All-unit discount



$$C^{(l)}(x) := c^{(l)} \cdot \max\{\alpha_j(x), \beta^{(l)}\}$$

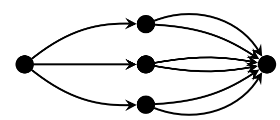


$$C(x) = \min_{l\in[L]} C^{(l)}(x)$$



	e	e'	e''
c	$c^{(l)}\beta^{(l)}$	0	0
c_i	0	$\alpha_{ij}c^{(l)}$	0
β_j	∞	∞	$\beta^{(l)}$

Minimum modeled by parallel gadgets



the pattern expanded network by the corresponding gadgets for their tariffs, we obtain the *tariff-expanded network* $G = (V, E)$ consisting of the nodes of the pattern-expanded network, the additional nodes introduced in the gadgets, and the edges introduced in the gadgets.

Each container of edge e has a capacity for every property. A solution to the container-based formulation of our model specifies for each edge e the (integer) number of containers $y(e)$ installed at e together with the edge flow values $x_i(e)$ for each commodity i. For each property, the capacity installed at e must be sufficient to transport the flow. More formally, recall that α_{ij} denotes the per unit extent of commodity i w.r.t. property j. Let $\beta_j(e)$ be the corresponding capacity of a container at edge e. Then the *capacity constraints*

$$\sum_{i\in K} \alpha_{ij} x_i(e) \le \beta_j(e) y(e) \quad \forall\, j \in P \tag{2}$$

must hold at every edge $e \in E$. Moreover, an upper bound $u(e)$ on the number of containers installed on an edge e may be specified.

In a feasible solution, the multicommodity flow x has to satisfy all demands. We extend the node balances introduced for the nodes in the pattern-expanded network by setting the balances for all nodes artificially introduced by tariff expansion to zero for each commodity. We thus obtain the *flow conservation constraints*

$$\sum_{e\in\delta^+(v)} x_i(e) - \sum_{e\in\delta^-(v)} x_i(e) = b_i(v) \quad \forall\, i \in K \tag{3}$$

that must be valid at every node $v \in V$ of the tariff-expanded network.

For each container installed at e, a fixed cost $c(e)$ has to be paid. Flow sent along e may furthermore incur a commodity-dependent linear cost $c_i(e)$ (which may naturally be used to model property-dependent linear costs as well). Thus, the total cost of a solution is

$$\sum_{e\in E}\left(c(e)y(e) + \sum_{i\in K} c_i(e)x_i(e)\right).$$

Putting all of this together, the fixed-charge multicommodity flow problem resulting from the container formulation can be directly formulated as an MIP.

$$\min \quad \left\{ \sum_{i\in K}\sum_{e\in E} c_i(e)x_i(e) + \sum_{e\in E} c(e)y(e) \right\}$$

$$\text{s.t.} \quad \sum_{e\in\delta^+(v)} x_i(e) - \sum_{e\in\delta^-(v)} x_i(e) = b_i(v) \quad \forall\, v \in V,\ i \in K,$$

$$\sum_{i\in K}\alpha_{ij}x_i(e)\le \beta_j(e)y(e) \quad \forall\, e\in E,\ j\in P,$$
$$y(e)\le u(e) \quad \forall\, e\in E,$$
$$x_i(e)\in\mathbb{R}_+,\ y(e)\in\mathbb{Z}_+ \quad \forall\, e\in E,\ i\in K.$$

Note that a flow in the tariff expanded network (i.e., on edges) can be transformed into a flow in the pattern expanded network (i.e., on transport relations) by setting $x(t)$ to be the amount of flow going from start(R) to end(R) through the gadget corresponding to t, which corresponds to the total amount shipped using this tariff.

The gadget of each tariff t will be designed to model its cost function C_t in the sense that the cost incurred by the flow in the gadget (in terms of required container capacity and linear costs) equals $C_t(x(t))$. Therefore, the total cost of the solution in the tariff-expanded network equals the cost of the flow in the pattern-expanded network.

2.4.2. Modeling Tariffs with Containers.

We now proceed to explain how containers can be used to accurately model the different types of transportation tariffs introduced in the previous section; see Table 1 for an overview of the more complex gadgets.

Modeling linear and piecewise constant costs. It is clear that both commodity-dependent linear costs and property-dependent piecewise constant costs are directly captured by the container concept. Linear costs are part of the definition, and piecewise constant tariff groups can be directly modeled by introducing a bundle of parallel edges, one for each tariff in the group. The container on each edge takes the capacity and cost of the corresponding tariff.

Modeling the maximum over multiple cost rates. To model the maximum over multiple cost rates we need to introduce *fractional containers* to the model; i.e., the variable $y(e)$ corresponding to the number of installed copies of such a container can be fractional. We use a single gadget edge for each tariff that corresponds to a maximum over multiple cost rates c_j with $j\in P$. We set the cost to $c(e)=1$ and the capacity $\beta_j(e)=1/c_j$ for each $j\in P$. Sending a flow of $x(e)$ through this edge requires $y(e)$ to be set to $\max_{j\in P}\{\alpha_j(x(e))/\beta_j(e)\}$, which is equal to the cost function by choice of $\beta_j(e)$. Note that introducing such fractional containers does not have a significant impact on the complexity of the model. For the sake of simplicity, we will assume throughout this work that all containers have to be installed in integral increments.

Modeling incremental discounts. Piecewise linear concave functions arising from incremental discount tariffs can be interpreted as the minimum of several affine linear functions. Again denoting the linear segments of the function by 0 to L with cost rates $c^{(l)}$ and break points $\beta^{(l)}$, we define

$$C^{(l)}(x) := c^{(l)}\alpha_j(x)+b^{(l)} \quad \text{with}$$
$$b^{(l)} := \sum_{k=0}^{l-1}(c^{(k)}-c^{(l)})(\beta_j^{(k+1)}-\beta_j^{(k)})$$

for $l\in[L]$. It is easy to verify that $C_t(x)=\min_{l\in[L]}C^{(l)}(x)$; see Table 1 for an illustration. We now introduce a gadget of $L+1$ parallel edges $e_0,\ldots,e_L$ with $c(e_l)=b^{(l)}$ and $c_i(e_l)=\alpha_{ij}c^{(l)}$. Sending flow along edge e_l incurs the cost $C^{(l)}$, and an optimal solution will always send flow along that edge, which achieves the minimum cost for the transported amount.

Modeling all-unit discounts. Note that functions of the form $c^{(l)}\cdot\max\{\alpha_j(x),\beta^{(l)}\}$ can be modeled by the following gadget; also see the corresponding figure in Table 1. We introduce a series-parallel graph, consisting of a single edge e followed in series by two parallel edges e' and e''. We set the fixed costs $c(e)=c^{(l)}\beta^{(l)}$ and $c(e')=c(e'')=0$ and the linear costs $c_i(e)=c_i(e'')=0$ and $c_i(e')=\alpha_{ij}c^{(l)}$ for all $i\in K$. Capacity $\beta_j(e'')$ is set to $\beta^{(l)}$; all other capacities are left infinite, and we let $u(e'')=1$ so that only one container can be installed on e'', and the number of containers remains unbounded for all other edges. Now, all-unit discount tariffs, which can be represented as the minimum of such functions, can be modeled by introducing several of these gadgets in parallel.

Remark. We close this section by pointing out two more general concepts that our model implicitly covers. First, the TTP model includes the possibility of omitting some holdover edges or even some transport edges of the base network in individual time slots, in order to model restricted operation times of transportation services or hubs. The second concept is abstract aspects of commodities, such as "needs cooling," "is hazardous," and similar features restricting the transportation. These can be modeled by introducing a corresponding property, letting the respective commodities receive a strictly positive extent in this property and accordingly adjusting container capacities.

3. Tariff-Selection Subproblem

Although containers constitute a versatile tool to model various transport tariffs, as described in §2.3, the use of elaborate gadgets significantly increases the number of edges in an instance of our model. Different solution algorithms may or may not be able to cope well with this challenge. In this section, we describe an approach to curb the effects of model blowup that results from tariff gadgets by encapsulating tariff selection decisions in a subordinate optimization problem, that we call the *tariff selection* (TS) subproblem. Although some of our algorithms for TTP introduced in §§4 and 5 will operate directly on tariff gadgets as introduced in §2.3, others will solve TS repeatedly, possibly very often for each transport relation, while computing a flow pattern for all commodities through the network.

In contrast to the global perspective of the TTP model, TS constitutes a local decision limited to a single transport relation $R \in \mathcal{R}$: Given a fixed vector $\bar{x}(R) \in \mathbb{R}_+^K$ of flow to be transported on R, it asks which transport tariffs should be selected and how the fixed demand should be distributed among selected tariffs to meet flow demand at a minimum cost. More formally, the problem TS for transport relation $R \in \mathcal{R}$ can be stated as

$$\begin{aligned} \min \quad & \sum_{t \in T(R)} C_t(x(t)) \\ \text{s.t.} \quad & \sum_{t \in T(R)} x_i(t) = \bar{x}_i(R) \quad \forall i \in K, \\ & x_i(t) \in \mathbb{R}_+ \quad \forall t \in T(R). \end{aligned}$$

A solution to TS comprises a vector $x(t) \in \mathbb{R}_+^K$ of multicommodity flow for each tariff $t \in T(R)$ such that their sum meets the total flow demand $\bar{x}(R)$. From a networkwide perspective, solving the union of the TS problems on all transport relations optimizes transport cost with respect to a given fixed multicommodity flow in the pattern-expanded network.

Depending on which of the five types of tariff cost functions introduced in §2.3 are present in TS, we employ different techniques to solve TS. In §3.1 we devise an MIP formulation for arbitrary combinations of tariff cost functions in TS. However, out of the different tariff cost functions, property-dependent piecewise constant costs stand out for a number of reasons. First, although they constitute the most elementary class of cost functions, in the presence of multiple tariffs of this type, determining an optimal tariff selection already is *NP*-hard (see Proposition 1). Second, it may be the tariff type occurring most frequently in logistics applications: Indeed, in the real-life data for our computational study in §6, many transport relations are equipped exclusively with piecewise constant tariffs. Therefore, §3.2 is devoted to theoretic and algorithmic insights into TS for this tariff type.

Elaborate algorithms for TTP, which we present in §4, solve TS as a subroutine very frequently. Because of its hardness and the demand for extremely short computation times, we develop fast heuristic algorithms for piecewise constant tariffs yielding only approximate solutions as an alternative to the exact MIP approach. In particular, we propose an efficient greedy algorithm for computing solutions of decent quality within a minimum of computation time in §3.2.1 and a *cost estimator* that, instead of a feasible solution, only outputs an estimate of the optimal cost of the given instance; see §3.2.2.

3.1. MIP for the General Case

The introduction of tariff gadgets in §2.4 enables us to naturally formulate and solve TS as a mixed-integer program. This versatile approach is especially suited when various tariff types occur together on a single transport relation, or when computational time is not a great issue, e.g., if flow paths for all commodities are already specified and TS only needs be solved once on each transport relation to optimize tariff choice. When each tariff $t \in T(R)$ is represented by a container gadget $(V(t), E(t))$, as detailed in §2.3, we denote with $E(R) := \bigcup_{t \in T(R)} E(t)$—respectively $V(R) := \bigcup_{t \in T(R)} V(t)$—the set of all edges—respectively nodes—that are introduced to model the tariff structure on transport relation R. TS for R can then be written as

$$\min \left\{ \sum_{e \in E(R)} c(e)y(e) + \sum_{i \in K} c_i(e)x_i(e) \right\}$$

$$\text{s.t.} \quad \sum_{e \in \delta^+(v)} x_i(e) - \sum_{e \in \delta^-(v)} x_i(e) = \begin{cases} \bar{x}_i(R) & \text{if } v = \text{start}(R) \\ -\bar{x}_i(R) & \text{if } v = \text{end}(R) \\ 0 & \text{otherwise} \end{cases} \quad \forall v \in V(R), \forall i \in K,$$

$$\sum_{i \in K} \alpha_{ij} x_i(e) \le \beta_j(e) y(e) \quad \forall e \in E(R), \forall j \in P,$$

$$y(e) \le u(e) \quad \forall e \in E(R),$$

$$y(e) \in \mathbb{Z}_+, \; x_i(e) \in \mathbb{R}_+ \quad \forall e \in E(R), \forall i \in K.$$

As this MIP represents TS only on one single transport relation, the MIP instances are rather small and can be solved near optimally in a reasonable time for matters of postoptimization.

3.2. Piecewise Constant Costs

When all tariffs on a transport relation are of the property-dependent piecewise constant type, the tariff-expanded transport relation is a bundle of parallel fixed-charge container edges. The MIP formulation of TS can be simplified to

$$\begin{aligned} \min \quad & \sum_{e \in E(R)} c(e)y(e) \\ \text{s.t.} \quad & \sum_{e \in E(R)} x_i(e) = \bar{x}_i(R) \quad \forall i \in K, \\ & \sum_{i \in K} \alpha_{ij} x_i(e) \le \beta_j(e) y(e) \quad \forall e \in E(R), \forall j \in P, \\ & y(e) \le u(e) \quad \forall e \in E(R), \\ & y(e) \in \mathbb{Z}_+, \quad x_i(e) \in \mathbb{R}_+ \quad \forall e \in E(R), \forall i \in K. \end{aligned}$$

It is not hard to see that solving TS to optimality remains *NP*-hard here, even for very restricted special cases. We give a straightforward reduction from the well-known unbounded knapsack problem, which is proven to be *NP*-hard (Lueker 1975), to TS instances with a single property, a single commodity, and no bounds on the container multiplicities.

Proposition 1. *Problem TS is NP-hard, even when restricted to instances with only piecewise constant cost functions, a single property, and a single commodity and unbounded multiplicities.*

Proof. In the single-commodity, single-property case, the above MIP reduces to $|E(R)|+1$ nontrivial constraints, and there remain three single parameters α_{ij}, $\bar{x}_i(R)$, and β_j, which we denote by α, $\bar{x}$, and β, respectively. Every feasible solution satisfies $\alpha\bar{x} \le \sum_{e\in E(R)} \beta(e)y(e)$, and conversely, if this inequality is satisfied, it is trivial to find feasible assignments $x(e)$. Hence, the MIP reduces in fact to a single nontrivial constraint.

An instance of the unbounded knapsack problem is given by a set of n items with values $v_1,\dots,v_n \in \mathbb{Z}_+$ and weights $w_1,\dots,w_n \in \mathbb{Z}_+$, a capacity $W \in \mathbb{Z}_+$, and a desired value $V \in \mathbb{Z}_+$. The task is to find numbers $z_1,\dots,z_n \in \mathbb{Z}_+$ such that $\sum_{i=1}^n w_i z_i \le W$ and $\sum_{i=1}^n v_i z_i \ge V$.

Given such an instance I_{UK} of the unbounded knapsack problem, we construct an instance I_{TS} of the above special case of TS as follows. First, for every item $i \in \{1,\dots,n\}$ of I_{UK}, define $u_i := \lceil W/w_i \rceil$ as the maximum number of items of type i in a feasible knapsack solution. Then, for each item $i \in \{1,\dots,n\}$, introduce a corresponding edge e_i with containers of fixed cost $c(e_i) = v_i$ and capacity $\beta(e_i) = w_i$. We set $\bar{x} = \sum_{i=1}^n w_i u_i - W$ and $\alpha = 1$.

We now argue that I_{UK} possesses a solution with value at least V if and only if I_{TS} can be solved with cost at most $\sum_{i=1}^n v_i u_i - V$. First assume there is a feasible solution z to I_{UK} with value at least V. We define $y(e_i) := u_i - z_i$ and observe that

$$\sum_{i=1}^n \beta(e_i)y(e_i) = \sum_{i=1}^n w_i(u_i - z_i) \ge \sum_{i=1}^n w_i u_i - W = \alpha\bar{x}$$

and

$$\sum_{i=1}^n c(e_i)y(e_i) = \sum_{i=1}^n v_i(u_i - z_i) \ge \sum_{i=1}^n v_i u_i - V.$$

We omit the converse of the argument, as it works analogously. □

3.2.1. Greedy Algorithm. In this section we present a generic greedy algorithm to heuristically solve instances of TS for piecewise constant cost functions. The inherent covering nature of TS—in the sense that we select containers to "cover" the capacity extents of a fixed flow vector $\bar{x}(R)$—motivates us to devise a generalization of the natural greedy approach to integer programs with nonnegative data, e.g., as studied by Dobson (1982).

The greedy algorithm for tariff selection repeatedly selects a "most efficient" container $e \in E(R)$ to cover portions of, or the whole remaining commodity demand $\bar{d}$, initialized by $\bar{d} := \bar{x}(R)$. Here, "efficiency" of a container is measured by the function $\texttt{Score}(e, \bar{d})$, which reflects the ratio between cost of container e and the portion of the demand $\bar{d}$ it covers. The selected container then is packed using the function $\texttt{Fill}(e, \bar{d})$, which returns a mix of commodities $\Delta \in \mathbb{R}_+^K$, with $\Delta_i \le \bar{d}_i$ for all $i \in K$ and $\alpha_j(\Delta) \le \beta_j(e)$ for all $j \in P$, to ensure "efficient" capacity use of the container. To speed up the algorithm we can assign the computed mix of commodities Δ multiple times to copies of the same container, as long as there is enough remaining demand $\bar{d}$ to assign Δ completely. Whenever the number $y(e)$ of selected containers reaches its upper bound $u(e)$, the container type e is removed from the set of available containers (we omit dealing explicitly with this case in our algorithms for better readability). The algorithm repeats until all demand is assigned, i.e., until $\bar{d}$ is reduced to zero. At some point in the algorithm there might be containers large enough to cover all remaining demand $\bar{d}$, whereas the $\texttt{Score}$ method still favors a smaller container that covers only fractions and leaves demand for the next step. In such situations it is advisable to consider both container types and to branch on the computed solution. The first branch completes the partial solution with a minimum cost container, that suffices to cover $\bar{d}$. The second branch proceeds with a container with best $\texttt{Score}$ value and iterates.

A formal listing of the greedy algorithm is given as Algorithm 1. To simplify notation we associate a multiset Y over $E(R)$ with a possible solution vector $y \in \mathbb{Z}_+^{E(R)}$ that contains $y(e)$ copies of container $e \in E(R)$ and denote with $c(Y)$ the respective selection cost $c(y) = \sum_{e\in E(R)} c(e)y(e)$.

Algorithm 1 (Greedy Algorithm For Tariff Selection)
Input: a TS subproblem on transport relation R with demand $\bar{x}(R)$
Output: assignment commodity vectors $x'(e) \in \mathbb{R}_+^K$, multiset Y' over $E(R)$

```
d̄ ← x̄(R);                  //remaining uncovered demand
x(e) ← 0, ∀e ∈ E(R);   Y ← ∅;
                            //current partial solution
x'(e) ← 0, ∀e ∈ E(R);   Y' ← ∅;
                       //current best complete solution
while there is uncovered demand d̄ do
  if there exists e_F = arg min_{e∈E(R): α(d̄)≤β(e)} c(e) then
                           //there is e_F that can store d̄
    if Y' = ∅ or c(Y ∪ e_F) < c(Y') then
                            //found new best solution?
      replace Y' with Y ∪ e_F and x'(e) with x(e),
        ∀e ∈ E(R);
      x'(e_F) ← x'(e_F) + d̄;
                            //update new best solution
  e_B ← arg max_{e∈E(R)} Score(e, d̄);
                         //pick most efficient container
```

$\Delta \leftarrow \mathsf{Fill}(e_B, \bar{d})$;
`//compute mix of commodities to assign`
$\mathsf{n} \leftarrow \lfloor \min_{i \in K:\, \Delta_i \neq 0} (\bar{d}_i / \Delta_i) \rfloor$;
`//compute multiplicity of assignment`
$Y \leftarrow Y \cup_{i=1,\cdots,\mathsf{n}} \{e_B\}$;
`//add container copies`
$x(e_B) \leftarrow x(e_B) + \mathsf{n} \cdot \Delta$;
`//update assigned commodities`
$\bar{d} \leftarrow \bar{d} - \mathsf{n} \cdot \Delta$;
`//compute remaining uncovered demand`
if $Y' \neq \emptyset$ *and* $c(Y) \geq c(Y')$ **then**
  return $x'(e), Y'$;
  `//complete solution dominates partial solution`

Implementation of `Score` *and* `Fill`*: A Two-Phase Greedy Algorithm.* Algorithm 1 uses two subprocedures called `Score` for estimating "container efficiency," and `Fill` for computing corresponding container packings. Both `Score` and `Fill` are based on a two-phase greedy algorithm that tries to pack a given container e by approximating the ray induced by the capacity vector $\beta(e)$. `Score` only executes the first phase of this algorithm and uses the resulting filling Δ to return the score $\sum_{j \in P} \alpha_j(\Delta)/c(e)$. Note that `Score` is executed far more frequently than `Fill`, so restricting to the first phase significantly saves computation time. Once a container is selected, `Fill` returns the refined filling derived by the second phase.

The outline of the two-phase algorithm is as follows. The first phase adds commodities that minimize the residual capacity of a container until one of the capacity constraints becomes tight or the demand of every commodity is depleted. Assuming that some commodity demands have already been added to Δ, let $\bar{\beta}(e)$ be the vector of residual capacities of this container w.r.t. Δ. For any given vector of commodities $\delta \in \mathbb{R}_+^K$, we denote the maximal fraction of Δ that can be feasibly assigned to a container with residual capacities $\bar{\beta}(e)$ by

$$\mathtt{linFrac}(\delta, \bar{\beta}(e)) := \min_{j \in P:\, \alpha_j(\delta) \neq 0} \frac{\bar{\beta}_j(e)}{\alpha_j(\delta)}.$$

Now the algorithm successively chooses a commodity i that minimizes the Euclidean norm of the vector of slacks after maximal feasible assignment of this commodity, i.e.,

$$i = \arg\min_{i' \in K} \left\| \bar{\beta}(e) - \min\{\mathtt{linFrac}(\bar{d}^{i'}, \bar{\beta}(e)), 1\} \cdot \alpha(\bar{d}^{i'}) \right\|,$$

where $\bar{d}^i$ is defined as $\bar{d}^i := (0, \ldots, \bar{d}_i, \ldots, 0)$ and adds this amount of commodity i to the current vector Δ. Phase 1 might incur an unnecessary amount of slack in some capacities because of the greedy choice of commodities. To improve this, Phase 2 minimizes slack by focusing on a good mix of assigned commodities.

It adjusts the current Δ to approximate the ray induced by the capacity vector $\beta(e)$ with a conic combination of property vectors α_i of the available commodities. More formally, we decompose the property space $\mathbb{R}^P = V(\beta(e)) + V(\beta(e))^\perp$ into the linear subspace $V(\beta(e))$ spanned by the capacity vector $\beta(e)$ and its orthogonal complement and consider for each commodity i the unique decomposition of its property vector $\alpha_i = v_i + u_i$ with $v_i \in V(\beta(e))$ and $u_i \in V(\beta(e))^\perp$. The current commodity mix $\Delta \in \mathbb{R}_+^K$ induces the property vector $\sum_{i \in K} \Delta_i \alpha_i = \sum_{i \in K} \Delta_i v_i + \sum_{i \in K} \Delta_i u_i \in \mathbb{R}_+^P$. Our goal of approximating the ray spanned by $\beta(e)$ corresponds to minimizing the orthogonal deviation $\|\sum \Delta_i u_i\|$. For commodity $l \in K$, we define $\lambda_l := \langle \sum \Delta_i u_i, u_l \rangle / \|u_l\|^2$. Note that $\lambda_l u_l$ corresponds to the projection of $\sum \Delta_i u_i$ on $V(u_l)$. If $\lambda_l < 0$, we augment Δ by $\min\{-\lambda_l, \bar{d}_l\}$ units of commodity l, which leads to a decrease of the orthogonal deviation. We iteratively augment Δ in this way until no additional improvement can be achieved by any commodity. Note that the resulting vector Δ might violate container capacities. We therefore scale Δ down to feasibility.

3.2.2. Cost Estimation by Covering Relaxation. In many situations where TS occurs as a subproblem in the course of an algorithm for TTP, it is not important to know which tariffs are utilized in a solution, just which cost is incurred. Examples include the shortest path type algorithms where the neighbors of some node are to be labeled with the cost of forwarding some flow to them. In these situations, the following covering relaxation can be used to obtain considerable speedups while still computing reasonable cost estimates. The relaxation is based on dropping the requirement of an exact assignment of the commodities to containers. Instead, we only require the chosen containers to cover the vector of aggregated properties $\bar{\alpha} := \alpha(\bar{x}(R))$ induced by the flow vector $\bar{x}(R)$. The result of this relaxation is the following covering problem (CR):

$$\begin{aligned} \min \quad & \sum_{e \in E(R)} c(e) y(e) \\ \text{s.t.} \quad & \sum_{e \in E(R)} y(e) \beta_j(e) \geq \bar{\alpha}_j \quad \forall j \in P, \\ & y(e) \in \mathbb{Z}_+ \quad \forall e \in E(R). \end{aligned}$$

We can heuristically solve this problem very efficiently by adjusting Algorithm 1 to directly operate on the property vector $\bar{\alpha}$; i.e., we reduce $\bar{\alpha}$ by $\beta(e)$ for each selected container copy e. An appropriate scoring function can be defined by $\mathtt{Score}(e, \bar{\alpha}) := (1/c(e)) \cdot \min_{j \in P} \{\beta_j(e)/\bar{\alpha}_j\}$. Note that a solution to the covering relaxation does not necessarily yield a feasible solution for the original TS problem. In fact, one can easily come up with counterexamples where the estimate obtained from CR is arbitrarily far away from the actual optimal solution value of TS. However, these examples are of rather artificial nature, including containers with zero capacity in certain properties.

4. Path-Based Local Search

We propose a local search procedure that employs local changes on a path decomposition of flow in the pattern expanded network using tariff-selection subroutines. As described in the introduction, there already are a number of local search heuristics available for solving capacitated network design problems. Adapting those methods to multiple capacities and nonbinary design variables does not suffice to cope with the large sizes occurring from practical application of our model. The precise replication of complex tariff structures leads to a drastically increased number of (mostly parallel) edges, which is further amplified by the cyclic expansion of the network (to give rough numbers, the tariff-expanded networks in our computational study have 250,000 edges on average, corresponding to a blow-up factor of 60 from an average of 4,000 edges for the base networks). This makes it very hard for heuristics that operate in the tariff-expanded network without knowledge of the tariff structure. Although most methods known from the literature either work directly on the design variables or reroute flow of a single commodity, our approach applies a neighborhood search that is based on path decomposition of flow in the pattern expanded network and reroutes multiple commodities simultaneously.

To obtain good initial solutions for the local search algorithm presented in §4.3, we also provide two successive shortest path-type algorithms, one that linearizes costs (SPLC) by estimating the per unit cost (§4.1) and one, denoted by SPTS, that uses a tariff-selection method for this purpose (§4.2). The first method was designed with an emphasis on speed and low memory requirement; the second is more accurate in cost estimation and is used as the central subroutine in our local search-improving moves.

We observed that our local search very well detects cost savings from splitting up flow sharing the same transport relation and rerouting it separately. In contrast, detecting potential savings from consolidating a diverse set of flow-carrying paths along a shared subpath is not well captured. Note that this effect may appear only after consolidating multiple paths—identifying such a set of paths is an algorithmically challenging task. To address this issue, we adapt the two path-based algorithms to encourage consolidation by (i) forbidding the direct source-sink-connections (which is well suited for our types of practical networks) in the SPLC heuristic and (ii) using a partial linearization technique for SPTS. Both refinements yield considerable improvements in solution quality of the local search procedure, as we will see in §6.

4.1. Shortest Paths with Linearized Costs

A straightforward idea for obtaining edge costs for a shortest path computation is estimating the per unit shipping cost on each edge in the tariff-expanded network by linearizing the fixed costs. This technique yields a highly efficient approach suited for solving even the largest occurring instances in a very small amount of time.

In each iteration, the algorithm chooses a commodity and finds a shortest path from a source to a sink with respect to edge weights $w \in \mathbb{R}_+^E$. Whenever the algorithm encounters an edge during the shortest path computation, the residual capacity for the chosen commodity on this edge is computed and the fixed cost for that edge is divided by this capacity to obtain a linear cost rate. To make this more precise, let $k \in K$ be the commodity that is currently being routed and (x, y) be the current (partial) solution to the capacitated network design formulation consisting of the flow $x \in \mathbb{R}_+^{K\times E}$ and design choices $y \in \mathbb{Z}_+^E$. Using the notation introduced in §2.4, we compute the *residual capacity* of edge $e \in E$ for commodity k provided by the $y(e)$ containers currently installed on the edge. This capacity is defined by

$$\rho(e) := \min_{j\in P} \frac{\beta_j(e)y(e) - \alpha_j(x(e))}{\alpha_{kj}}.$$

If there is a positive residual capacity $\rho(e) > 0$, we set the capacity $r(e) := \rho(e)$ and the weight $w(e) := c_k(e)$, only considering the linear cost for shipping commodity k along e. If no residual capacity is left, i.e., $\rho(e) = 0$, then an additional container can be installed on e if $y(e) < u(e)$. In this case, we set

$$r(e) := \min_{j\in P} \frac{\beta_j(e)}{\alpha_{kj}} \quad \text{and} \quad w(e) := c_k(e) + \frac{c(e)}{r(e)}.$$

Otherwise, if $y(e) = u(e)$, we set $r(e) := 0$ and $w(e) := \infty$.

Once a shortest path P from a source to a sink of commodity k with respect to the weights w is found, the *bottleneck capacity* $r := \min_{e\in P} r(e)$ is determined and r units of commodity k are sent along the path. Note that all of the above computations can be carried out efficiently, and instead of updating weights and capacities of all edges in each step, these values are calculated on demand and updated when necessary. Technically, we flag those variables $r(e)$ and $w(e)$ as "invalid" that have to be recomputed at the next encounter of edge e. A listing of SPLC is given as Algorithm 2.

Algorithm 2 (Successive Shortest Path Algorithm with Linearized Costs (SPLC))

```
Initialize x = 0, y = 0.
for each commodity i ∈ K do
    Invalidate r(e) and w(e) for all e ∈ E.
    while there is a source s of i with remaining supply do
        Find path P in G from s to a sink t with
          ∑_{e∈P} w(e) minimum, updating the values
          of r(e) and w(e) on-demand when the
          previous value has been invalidated.
        Augment x along P by min_{e∈P} r(e) units
          of commodity i, adjust y accordingly.
        Invalidate r(e) and w(e) for all e ∈ P.
```

The linearization procedure assumes optimal utilization of container capacities in the resulting flow pattern and thus favors large containers with low per unit cost rates. Since this high utilization is not always attained, the linearization often leads to suboptimal tariff choices on transport relations. The effect can be compensated by optimizing the tariff selection on each transport relation a posteriori with a tariff selection method described in §3.

Consolidation by Forbidding Direct Connections (SPLC-F). The SPLC heuristic favors large containers with low per unit costs and prefers direct connections, as single detours cannot yield lower per unit costs. A simple approach for encouraging consolidation when costs are just linearized is to forbid all direct connections between sources and sinks of the same commodity during the construction of the initial solution. By doing so, hubs and common paths are automatically used. Unnecessary detours can be easily identified and corrected by improving moves of the local search procedure.

4.2. Shortest Paths with Tariff Selection

The rather imprecise estimation of the actual transportation cost achieved by the linearization approach presented in the previous section might lead to weak choices of paths when routing the commodities. We thus propose a second strategy that employs tariff selection algorithms during the shortest path search. Although this more sophisticated approach requires more computational effort, it still turns out to be efficient while at the same time providing several possibilities for adjustments.

Since tariff-selection methods require as input the amount of flow to be routed, these flow values $\Delta \in \mathbb{R}_+^K$ have to be determined *before* the shortest paths computation. We implement this a priori flow computation efficiently by identifying source-sink-pairs such that the possible transport volume from source to sink is maximum (w.r.t. a weighted combination of the property extents).

More formally, for each ordered pair of nodes (s, t) in the pattern expanded network, let

$$\Delta_k(s, t) := \big(\min\{b_k(s), -b_k(t)\}\big)^+$$

for $k \in K$, and let $w \in \mathbb{R}_+^P$ be the weight function, given as a parameter to the heuristic. Then source s and sink t are chosen such that $\sum_{j\in P} w_j \alpha_j(\Delta(s, t))$ is maximum.

During the shortest path computation, edge weights have to be evaluated too often to solve the tariff selection problem to optimality every time. In fact, it is sufficient to only estimate the cost using the estimator presented in §3, whereas the actual tariff assignment can be determined at the end of the solution process from the flow values on the transport relations in the pattern expanded network using an exact method. A listing of SPTS is given as Algorithm 3.

Algorithm 3 (Successive Shortest Path Algorithm with Tariff Selection (SPTS))

```
Initialize x = 0.
while not all demand has been satisfied do
    Let s, t ∈ 𝒱 such that ∑_{j∈P} w_j α_j(Δ(s, t)) is
      maximum.
    Compute shortest path P in ◁ from s to t w.r.t. c̃,
      where c̃(R) is the estimated cost for augmenting
      the current flow x(R) by Δ(s, t) on transport
      relation R.
    Augment x along P by Δ(s, t).
Compute a flow in the tariff expanded network of
same value as x using a tariff selection method.
```

Consolidation by Partial Cost Linearization (SPTS-L). Cost computation based on tariff selection allows for a more sophisticated approach to encourage consolidation by taking into account the unrouted demand. We linearize costs at inter-hub and source-hub (if there are fewer sources) or hub-sink (if there are fewer sinks) connections in the following way: Let $\Delta^+ \in \mathbb{R}_+^K$ be the sum of all supply not yet routed in the current solution, and let $M := \min_j((\sum_i \alpha_{ij}\Delta_i^+)/(\sum_i \alpha_{ij}\Delta_i))$. For each available tariff t on a transport relation, we now compute the cost $C_t(\Delta^+)$ for routing Δ^+ and divide it by M to obtain an edge cost that anticipates future consolidation on this transport relation.

4.3. Path-Based Local Search

In the following we introduce a local search algorithm that reroutes flow along paths with the aim of improving feasible solutions. Before we describe the procedure in detail, we briefly introduce the notion of flow decomposition.

A well-known result from network flow theory states that any feasible flow in a network can be decomposed into flow on paths from sources to sinks (and cycles, which, however, can immediately be removed from the solution in our case). A *flow-carrying path* is a tuple

(P, Δ_P), where P is a sequence of transport relations $R_1, \ldots, R_m$ such that $\text{start}(R_{i+1}) = \text{end}(R_i)$ and $\Delta_P \in \mathbb{R}_+^K$ is a multi-commodity flow vector specifying the amount of flow sent along the path. A *path decomposition* of a flow x is a collection of flow-carrying paths $\mathcal{P}$ such that $x(R) = \sum_{P \in \mathcal{P}:\, R \in P} \Delta_P$.

The local search algorithm maintains a path decomposition of the flow of the current solution. It moves from one solution to another by replacing one or multiple paths of the decomposition with paths of lower cost. The general outline of an improving move is the following: when removing a path (P, Δ_P) from the solution, for each transport relation R of the path, $x(R)$ is decreased by Δ_P and the tariff selection of R is adapted accordingly, using the greedy tariff selection heuristic presented in §3. After removing a set of paths, the resulting partial solution is completed again by computing new paths using the SPTS heuristic introduced in §4.2. The move is accepted if the total cost of the solution decreases, and is reverted otherwise.

We implemented two variants of improving moves: Type A moves simply to remove a single path at a time. This way, only small amounts of flow are rerouted in one move and the assignment of sources to sinks is left unaffected. In contrast, Type B moves consider groups of paths sharing the same transport relation. All flow passing this transport relation is removed and routed anew, which means that multiple paths can be replaced at once and the assignment of sources to sinks might be altered.

Our local search algorithm now performs improving moves in alternating phases of Types A and B. This allows us to recompute the path decomposition at the beginning of each phase, adapted to the type of movement.

In both cases paths are constructed in a depth-first search (DFS) manner: At a node in the DFS tree for each incident edge R we compute the maximal flow vector $\Delta(R)$ that could be assigned to a path proceeding on that edge and choose an edge greedily so as to maximize a suitably defined weight function of that flow vector. For Type A phases, the DFS starts at a source and continues along the edge that maximizes a weighted combination of the properties of $\Delta(R)$. In contrast, the decomposition for Type B phases facilitates a bidirectional DFS starting at heavily used transport relations and chooses edges that maximize the savings resulting from reducing their flow. In both cases, because of flow conservation, we either close cycles (which can immediately be removed from the solution) or find a source-sink path, which we add to the path decomposition.

The two phases alternate repeatedly until the relative improvement achieved by both falls below a specified value or the time limit is reached. At the end of the procedure, a final improvement phase is conducted by identifying and eliminating weakly utilized containers in the tariff-expanded network and again rerouting the corresponding flow using a variation of Type B moves.

5. MIP-Based Approaches

In this section, we discuss MIP techniques that supplement the combinatorial heuristics presented in the previous section, not only yielding high-quality solutions but also providing lower bounds for assessing this quality. The plain MIP formulation presented in §2.4 is not suited for solving reasonably sized real-world networks since they involve too many variables and constraints. We propose an aggregated formulation that considerably reduces model size and still yields good dual bounds (§5.1). We then combine this with efficient preprocessing techniques to tighten the relaxation (§5.2). In §5.3, we use solutions to the LP relaxation of this strengthened aggregated formulation as initial solutions for our local search. Finally, a postprocessing step that improves solution quality is presented in §5.4. During this step, tariff selection decisions are locally optimized on all transport relations that connect a given pair of nodes in different slots of the pattern-expanded network.

Besides strengthening the MIP formulation, a promising approach to deal with multicommodity capacitated network flow problems is to use a Benders decomposition; see, e.g., Costa, Cordeau, and Gendron (2009). Preliminary runs with a Benders decomposition combined with heuristics and adding additionally valid inequalties implemented in SCIP 2.0 suffered from slow solving times. Interestingly, the subproblems (multicommodity multidimensional flow problems) solved by CPLEX turned out to be the bottleneck. In fact, numerical instability results from high variance between large and small coefficients in our practical instances in conjunction with inexact dual values inherent to Benders decomposition. Experiments with warm starts in the subproblem solving procedure and other techniques did not work out on our large-scale tariff and pattern-expanded networks. To be precise, CPLEX tries several Markowitz thresholds and tries to repair basis singularities. We point out that on small instances our Benders implementation works well, but it seems to be the large instances that induce a huge amount of Benders cuts together with their widely varying coefficients and long LP solving times. We leave it to future research to determine how to incorporate multidimensional capacities into combinatorial approaches similar to Costa, Cordeau, and Gendron (2009).

5.1. Tariff-Aggregated MIP

As mentioned above, the plain MIP formulation suffers from huge memory requirements. In particular, the introduction of tariff gadgets results in a tremendous number of—mostly parallel—edges. We make use of this parallel structure and propose an aggregated formulation that still reflects the original tariff structures while significantly reducing the number of flow variables and capacity constraints. The aggregation is set up as follows. For each pair of nodes $v, w \in V$ let $E(v, w)$ be the set of edges from v to w in the tariff-expanded network. For each $i \in K$, we replace the flow variables $x_i(e)$ of the edges $e \in E(v, w)$ with a single flow variable $x_i(v, w) \in \mathbb{R}_+^K$. For each $j \in P$, we replace the capacity constraints of the edges in $E(v, w)$ w.r.t. j by a single constraint

$$\sum_{i \in K} \alpha_{ij} x_i(v, w) \le \sum_{e \in E(v, w)} \beta_j(e) y(e).$$

Clearly, the resulting MIP is a relaxation of the original TTP instance, as we can construct a feasible solution of the relaxation from a feasible solution of the original formulation by setting $x_i(v, w) := \sum_{e \in E(v, w)} x_i(e)$ and adopting the values of all design variables. Conversely, each solution of the relaxation induces a flow on the transport relations of the pattern-expanded network. These flow values yield a tariff-selection subproblem on each transport relation (see §3). Computational experiments on practical instances reveal that by applying a tariff-selection heuristic on each relation, we can derive feasible solutions of the original model with a minimal increase in cost. In contrast, given the typically high number of parallel edges between each pair of nodes in TTP instances (20 on average in our test sets), the aggregation drastically reduces the number of variables and constraints, resulting in a considerable boost in effectiveness of branch and bound solvers.

5.2. Preprocessing

Although tariff aggregation helps reduce problem sizes, the considered MIPs still suffer from numeric instability and weak lower bounds. We address these issues with two preprocessing steps that can be applied to the aggregated formulation.

5.2.1. Strengthened Container Inequalities. As already discussed in §1.3.3, MIP formulations of capacitated network design problems can be considerably strengthened by adding valid inequalities. Among the valid inequalities used in the literature are strong capacity and minimum cardinality inequalities. The natural extensions of these inequalities to TTP, however, did not turn out to be very effective for the instances in our computational study. Instead, we propose a method to bound the total extent of capacity used within individual containers. Before we describe these *strengthened container inequalities* in detail, we give some reasons for the failure of the known inequalities mentioned above.

Strong capacity inequalities state that $x_i(e) \le b_i y(e)$ for all $i \in K$ and all $e \in E$, where $b_i := \sum_{v \in V: b_i(v) > 0} b_i(v)$ is the total demand of commodity i. Although Chouman, Crainic, and Gendron (2011) report on the positive impact of strong capacity inequalities on the integrality gap in their computational experiments, it is also easy to see that the strong capacity inequality for commodity i at edge e can only strengthen the original formulation if $\beta_j(e) > \alpha_{ij} b_i$ for all $j \in P$. In typical TTP instances, total demands within the network are much larger than individual transport capacities and the inequalities remained mostly ineffective.

Minimum cardinality inequalities require the number of containers installed on a cut induced by a set of nodes $S \subset V$ to be at least as large as the minimum number of containers required to transport the excessive demand $(\sum_{v \in S} b_i(v))_{i \in K}^+$ within S across the cut. As already observed by Chouman, Crainic, and Gendron (2011), these inequalities are weak if the magnitudes of the capacities vary widely, as is typically the case for logistics tariffs that are modeled within TTP instances. Their suggested improvements cannot be applied in our case as their model contains only binary design variables, whereas ours are integer. In the following, however, we show how to strengthen our capacity inequalities using similar ideas.

Solutions to the LP relaxation of TTP provide weak lower bounds for the following reason: When considering a flow-carrying transport relation, LP solutions tend to set the variable of the largest container to the minimal fraction needed to grant capacities for the flow on this transport relation. These fractions are unfortunately very small, which means that they do not reflect the cost that would be incurred in an integer solution. The idea is to restrict container capacities without affecting the cost of an optimal integer solution. This is possible, if, for a given transport relation $R \in \mathcal{R}$, an upper bound $\Gamma(R)$ on the flow $x(R)$ in any optimal solution is known. Useful upper bounds can be derived for transport relations incident to node sets $S \subset V$ with either $\delta^+(S) = \varnothing$ or $\delta^-(S) = \varnothing$. Given an upper bound $\Gamma(R)$, we can replace for every $e \in E(R)$ and every $j \in P$ the capacity $\beta_j(e)$ by $\beta_j(e) - s_j$, where s_j is the result of solving

$$\begin{aligned} \min \quad & s_j \\ \text{s.t.} \quad & \sum_{i \in K} \alpha_{ij'} x_i(e) + s_{j'} = \beta_{j'}(e) \quad \forall j' \in P, \\ & 0 \le x_i(e) \le \Gamma_i(R) \quad \forall i \in K, \\ & s_{j'} \ge 0 \quad \forall j' \in P. \end{aligned}$$

In a preprocessing routine we solve these linear programs for each property j of each fixed-charge container e on each transport relation R for which reasonable upper bounds $\Gamma(R)$ can be computed.

5.2.2. Commodity Scaling. We could observe numerical difficulties while solving LP relaxations of large instances: the LP solving steps suffer from basis singularities and sometimes even numerical infeasibility. One reason for these difficulties lies in the diversity of properties for different commodities. The capacity inequalities involve many flow variables with property coefficients varying in magnitudes of 10^6 for our test instances. Nonetheless, because flow variables are fractional in our model, we can apply the following scaling steps. For each commodity $i \in K$ we determine a scaling factor $s_i > 0$ and obtain scaled values $\tilde{b}_i(v)$ and $\tilde{\alpha}_{ij}$, defined by

$$\tilde{b}_i(v) := b_i(v)/s_i \quad \text{and} \quad \tilde{\alpha}_{ij} := s_i \alpha_{ij} \quad \text{for each } j \in P.$$

The scaled problem instance is equivalent to the nonscaled one in the sense that feasible flow values $\tilde{x}_i(e)$ obtained for the scaled problem can be scaled back to obtain feasible flow values $x_i(e) = s_i \tilde{x}_i(e)$ for the original problem. We chose the scaling factors s_i for each commodity in such a way that among the resulting coefficients $\tilde{\alpha}_{ij}$, $j \in P$ the smallest such coefficient has the magnitude 10^{-1}. The improved numeric stability of the constraint system significantly speeds up the LP solution process.

5.3. Initial Solutions for Local Search from Aggregated LP Relaxation

In §4, we discussed the importance of properly chosen initial solutions for the local search procedure and devised two ways to encourage consolidation of flow during the construction of the initial solution by shortest path type algorithms. Alternatively, we can obtain initial solutions from the LP relaxation of the aggregated MIP formulation by applying tariff-selection heuristics to the multicommodity flow in the pattern expanded network induced by the aggregated LP solution.

Notice that in this case, strengthening container inequalities as described above also encourages consolidation in the solution process. In fact, the effect of the strengthened inequalities is strongest on edges that are reachable from a few sources or sinks only (such as direct source-sink connections). This implicitly encourages flow to take detours on non-source-sink paths, where weaker container inequalities permit lower costs in the LP relaxation. Since inappropriately consolidated flow can be efficiently disaggregated by the local search algorithm, initial solutions constructed from the LP relaxation lead to high-quality final solutions, as we shall see in §6.

5.4. Pattern-Optimization Subproblem

In the tariff-selection subproblem considered in §3, we fixed the amount of flow passing a given transport relation and optimized the tariff selection with respect to this given flow value. This idea can be extended by considering all transport relations that connect a given pair of nodes in different slots of the pattern-expanded network. More formally, for some node $v \in B$ in the base network and a cycle length F, let $v_1, \ldots, v_F$ be the copies of node v created in the pattern expansion step, with $v_i \in V(B_i)$ for $i \in \{1, \ldots, F\}$. We consider the *pattern-optimization subproblem* induced by a fixed pair of nodes s, $t \in B$ and therefore define

$$V(s,t) := \bigcup_{i=1}^{F} \{s_i, t_i\},$$

$$\mathcal{R}(s,t) := \{R \in \mathcal{R}\colon\ \text{start}(R), \text{end}(R) \in V(s,t)\}.$$

Given a solution to the whole TTP instance with flow values $(\bar{x}(R))$, $R \in \mathcal{R}$, we consider a locally restricted instance of TTP, fixing the flow values on all transport relations $\mathcal{R} \setminus \mathcal{R}(s,t)$ and optimizing the flow $(x(R))_{R \in \mathcal{R}(s,t)}$ in the subnetwork induced by the copies of s and t; i.e.,

$$\begin{aligned}
\min \quad & \sum_{R \in \mathcal{R}(s,t)} \sum_{t \in T(R)} C_t(x(t)) \\
\text{s.t.} \quad & \sum_{R \in \delta^+_{\mathcal{R}(s,t)}(v)} x_i(R) - \sum_{R \in \delta^-_{\mathcal{R}(s,t)}(v)} x_i(R) = \bar{b}_i(v) \quad \forall v \in \mathcal{V}(s,t),\ \forall i \in K, \\
& \sum_{t \in T(R)} x_i(t) = x_i(R) \quad \forall R \in \mathcal{R}(s,t),\ \forall i \in K, \\
& x(t) \geq 0 \quad \forall t \in T(R),\ \forall R \in \mathcal{R}(s,t),
\end{aligned}$$

where $\bar{b}(v) := \sum_{R \in \delta^+_{\mathcal{R}(s,t)}(v)} \bar{x}(R) - \sum_{R \in \delta^-_{\mathcal{R}(s,t)}(v)} \bar{x}(R)$. Using tariff gadgets, this restricted instance of TTP can be formulated as a mixed-integer program. It contains only a small fraction of the decision variables present in the whole instance. In fact, restricted instances can be solved to near optimality very quickly using a standard MIP solver. We thus iteratively optimize these subproblems arising for all pairs of adjacent nodes with flow carrying transport relations between them.

Note that in contrast to the tariff-selection subproblem, solving the pattern-optimization subproblem for one pair of nodes may affect the subproblem of other, nondisjoint pairs of nodes, as holdover edges of a common node appear in each of the problems as variables. Consequently, the order of the node pairs considered plays an important role. We order the node pairs nonincreasingly with respect to a weighted combination of the property extents of the total flow in the subnetwork affected by the pattern optimization of each pair (s,t), i.e., $\sum_{j \in P} w_j \alpha_j (\sum_{R \in \mathcal{R}(s,t)} \bar{x}(R))$, using the same weights $w \in \mathbb{R}^P_+$ as provided for local search and SPTS heuristic. This reflects the optimization potential of the corresponding node pair and leads to an "important pairs first" order, which is also useful when the pattern optimization process is not carried out on all node pairs because of time constraints.

6. Computational Study

We verify the TTP model and the algorithmic approaches presented in the preceding sections by conducting a computational study based on real-world data provided by our project partner, 4flow AG, a logistics consultancy company serving small, medium-size, and global customers from a broad spectrum of industries. We also compare our heuristics and MIP-based approaches with a reference solution obtained from 4flow AG.

6.1. Instance Sets

The benchmark library consists of 145 instances aggregated from four recent and ongoing customer projects in three different industries (Auto1, Auto2, Chemical, and Retail). All base networks correspond to European supply chains in which goods are transported according to FTL or LTL tariffs. These networks share a layered graph structure. More specifically, the nodes of the base network are partitioned into an ordered set of layers, with the lowest layer containing all sources and the highest layer containing all sinks. In addition, there is a fixed number (varying from one to three) of intermediate hub layers. There is a transport relation between every pair of nodes from distinct layers, directed toward the higher layer. However, transport relations within the same layer are not present. Pattern expansion has been conducted with a cycle length of six slots—one slot corresponds to two months of a year. All tariffs are of piecewise constant type, depending on the same two properties (mass and volume) in every instance.

The automotive instances represent production networks with a high number of sources and a low number of sinks; the chemical industry and retail sets are based on distribution networks with a high number of sinks but only a few sources. Table 2 shows the average values of key parameters of the instances within each set: the first three columns contain the number of sources, sinks, and hubs in the base network, followed by the number of commodities (comm.), and the number of edges in the base network, pattern-expanded network and tariff-expanded network.

For future research, the instance library will be available on request after signing a contract of data confidentiality. For more information, please contact one of the authors.

6.2. Algorithms and Implementation Details

We implemented and tested different variants of the algorithms presented in §§4 and 5 to determine good parameter settings and combinations. In long-term planning, running time plays a minor role and the fine-tuned aggregated MIP formulation, combined with the path-based local search and pattern optimization with generous time limits, can be used. To enable the evaluation of multiple scenarios, our industrial partner set a time limit of 30 minutes. For this case, we also provide test results of approaches designed for time efficiency without sacrificing too much solution quality.

Overall, the following algorithms were tested on all 145 instances of the benchmark library. The first two algorithms correspond to MIP approaches and the last four are local search procedures that are named according to the algorithm that delivers the initial solution for local search.

`AMIP-H`: Aggregated MIP with integrated local search (see §6.2.1).

`MIP`: Plain MIP formulation for comparison purposes, see §2.

`ALP`: LP relaxation of aggregated MIP formulation (see §5.3).

`SPLC`: Shortest path heuristic with linearized cost (see §4.1).

`SPLC-F`: same as `SPLC`, but with forbidden direct connections (see §4.2).

`SPTS-L`: Shortest path heuristic with tariff selection (cost estimator) and partial linearization (see §4.2).

All algorithms have been implemented in C++ and compiled with gcc 4.5.0 on openSUSE 11.3 Linux with kernel 2.6.32.19-0.2. Computations have been performed on cluster nodes with two Dual-Core Opteron 2218 processors (2.6 GHz, 64 bit) and 16 GB of memory using CPLEX 12.1 for MIPs and LPs. Since the heuristic approaches have not been adapted to support concurrency, we limited the number of threads for the CPLEX solver to one to ensure comparability of the results.

In §6.2.1 we elaborate on the interplay of the MIP and the local search heuristic; the detailed settings for the variants of local search procedures are presented in §6.2.2.

6.2.1. Branch and Bound Frameworks. Our tests involved different MIP formulations, which we implemented in CPLEX. We tested the plain MIP formulation

Table 2 Average Sizes of the Instances per Set

Set (No. of instances)	No. of nodes in base network: No. of source	No. of sinks	No. of hubs	No. of comm.	No. of edges: Base	Pattern exp.	Tariff exp.
Auto1 (36)	35	6	7	162	335	2,296	76,653
Auto2 (18)	34	3	4	117	186	1,364	29,264
Chemical (50)	7	244	19	101	6,601	41,222	239,238
Retail (41)	4	177	26	307	5,665	35,229	511,064

(MIP) for a direct comparison with our algorithms as well as the aggregated MIP formulation that includes the preprocessing methods described in §5.2 and callbacks to our heuristics. The resulting algorithm is denoted by AMIP-H and details of the implementation are given below. To obtain reasonably tight lower bounds, we also ran the aggregated MIP formulation without heuristic callbacks (AMIP-B). We invoked a time limit of two hours for the branch and bound process and an extra time of one hour for applying local search and pattern optimization.

When solving the aggregated and preprocessed MIP formulation from §5 with a branch and bound framework, we apply the local search and pattern optimization procedures throughout search on integer solutions as well as fractional LP solutions obtained in a node of the branch and bound tree. These solutions induce a flow on the transport relations of the pattern-expanded network. This flow can be turned into a feasible TTP solution by solving the tariff-selection subproblem on each transport relation (see §5.1). We further improve this solution by applying the local search heuristic and pattern optimization with a time limit of 300 seconds.

As this procedure incurs a significant computational effort, we require at least 1,500 branch and bound nodes to be processed between two successive calls of the heuristics. Furthermore, we use the cost estimator presented in §3.2.2 to evaluate the potential of a given LP solution to improve on the current best solution: only if the estimated total cost is within 8% of the best known solution do we compute the corresponding TTP solution. We also apply the procedure to all integer solutions found by the MIP solver.

6.2.2. Local Search Procedures. We tested the local search algorithm described in §4.3 using initial solutions constructed by the aforementioned heuristics. The current tariff selection on the transport relations was then further improved using the exact MIP formulation, as described in §3. Finally, pattern optimization was performed on the returned solution using the nonaggregated formulation.

Computation time of the starting heuristics ALP, SPLC and SPLC-F turned out to be almost negligible, and we invoked a total solution time of 30 minutes (including pattern optimization) in this case. Unfortunately, the more sophisticated SPTS-L solver turned out to cause considerably more computational effort. Here we invoked the same time limits as for the branch and bound approaches. Recall that for fine-tuning the path-decomposition of the local search procedure and the SPTS heuristic an additional parameter is specified—a weight function on the properties of the model that reflects the importance of properties. For the benchmark instance set, mass occurs to be the dominant property. We thus choose the weight function to be an indicator function on mass.

Table 3 **Average Improvement of the Lower Bound Compared to MIP**

Solver	Auto1 (31/36)	Auto2 (18/18)	Chemical (48/50)	Retail (30/41)	All (127/145)
ALP	−17.81	−6.87	−21.61	−10.44	−15.96
AMIP-B	17.48	0.61	12.38	4.84	10.18

Note. The number of instances handled by MIP/number of all instances per set is shown in parentheses.

6.3. Results

We now elaborate on the results of our computational experiments, starting with the effect of aggregation on the lower bounds. We then analyze solution quality and the impact of local search initial solutions and pattern optimization. We close by comparing our approach to a reference solution on an additional instance.

6.3.1. Influence of Aggregation on Lower Bounds. We investigate the improvement on lower bounds achieved by the aggregation and our preprocessing techniques against the plain MIP formulation in Table 3. In fact, we observed that especially for the large instances, MIP suffers from numerical instabilities and degeneracy that lead to solving times of thousands of seconds for the root relaxation. In some cases, the initial cut generation rounds for the root relaxation did not terminate within given time limits. In turn, the efficiency of initial cuts greatly benefits from our preprocessing techniques—fewer cuts achieve a much better lower bound here.

Not surprisingly, the lower bounds derived by the strengthened aggregated LP (ALP) are of low quality, with a gap of more than 15% on average toward the value obtained by MIP. In a set-by-set comparison, the AMIP-B method achieves an average improvement over MIP of more than 10% and up to 17% on average on set Auto1, whereas MIP is only competitive on the comparatively small instances of the Auto2 set. Apparently, the loss in tightness caused by the aggregation is more than compensated by the boost in efficiency of the branch and bound procedure achieved by the smaller size of the formulation and its increased numerical stability. An overview over the lower bounds for all instances can be found in the online appendix (available as supplemental material at http://dx.doi.org/10.1287/trsc.2014.0541).

6.3.2. Quality of Solutions. Table 4 shows the gaps of the computed solutions to the lower bound computed by AMIP-B. Throughout the automotive and retail instance sets, the solution quality is within single-digit average gaps to the lower bounds. The local search with LP starting solution and the AMIP-H framework provide the best solution quality, whereas the performance of approaches with path-based initial solutions is weaker and varies, depending on the instance set. We infer that the more holistic LP approach captures

Table 4 Average Gaps to Best Known Lower Bound in %

Solver	Auto1 (36)	Auto2 (18)	Chemical (50)	Retail (41)	All (145)
MIP	9.09 (1)	2.35 (3)	29.18 (0)	13.33 (1)	16.38 (5)
ALP	6.22 (24)	2.63 (0)	13.92 (17)	4.74 (40)	8.01 (81)
AMIP-H	6.12 (26)	1.26 (16)	14.61 (24)	4.75 (38)	8.06 (104)
SPLC	6.51 (17)	5.07 (0)	23.54 (0)	4.75 (37)	11.71 (54)
SPLC-F	6.90 (11)	3.65 (0)	18.08 (9)	10.70 (27)	11.43 (47)
SPTS-L	6.57 (19)	4.15 (0)	19.44 (1)	4.74 (39)	10.19 (59)

Note. Number of achieved best solutions is shown in parentheses.

the multicommodity flow nature of our problem better than the iterative path approaches.

AMIP-H attains near optimality on Auto2, outperforming ALP on this set. Apparently, the small instance sizes in this set benefit the branch and bound process.

The gaps are considerably weaker on the instances of the Chemical set. The instances of this set are much bigger w.r.t. the number of edges and sinks in the base network than those from the other sets, which presumably also affects the MIP framework's ability to produce tight lower bounds.

6.3.3. Performance of Local Search and Impact of Initial Solutions. The results in Table 4 and Figure 2 show that the choice of the initial solution clearly affects the performance of the local search procedure. In fact, in many instances, the initially expensive flow patterns of the consolidation enforcing heuristics lead to better final solutions than those obtained from solutions with low consolidation provided by SPLC for comparison. However, the effectiveness of the combinatorial starting heuristics strongly depends on the structure and size of the instance. In contrast, ALP consistently shows the best results, on par with the AMIP-H framework (which takes considerably more computational effort).

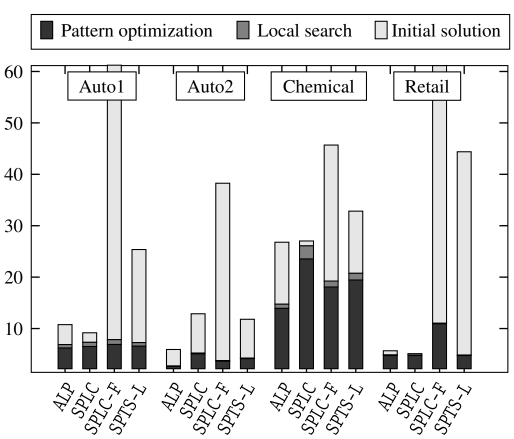


Figure 2 Gaps Achieved with Postprocessing in Percent
Note. The percentage gap to best known lower bound of fast solvers (time limit 1,800 seconds) for the initial solution, after local search and pattern optimization are shown—with initial solutions by SPLC-F achieving 113% on average in Auto1 and 253% in Retail.

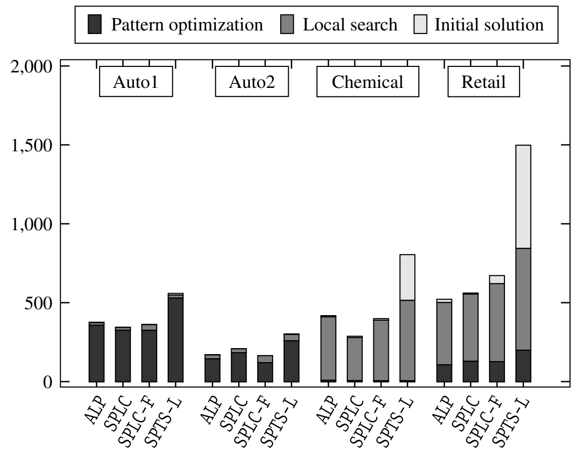


Figure 3 Running Times of Postprocessing
Note. The table shows shifted geometric means of running times as shares of 2,000 seconds for the local search combined with different algorithms to compute initial solutions with pattern optimization.

6.3.4. Impact of Pattern Optimization. Figure 2 reveals that the effect of pattern optimization is rather weak on the sets Auto2 and Retail, although its share of the computation times is significant (Figure 3). The picture is considerably different, however, for the instances of the Chemical set. Here, computation times are reduced to a minimum, whereas the improvement of solution quality because of pattern optimization is significantly higher. This better performance can be explained by the less granular tariff structure in this instance set, resulting in smaller subproblems while at the same time increasing the importance of temporal consolidation.

6.3.5. Purely Combinatorial Heuristics. To provide solutions independent of third-party software and licenses, we also evaluated purely combinatorial variants of the local search heuristic with path-based initial solutions: After replacing MIP-based tariff selection algorithms with greedy heuristics and omitting pattern optimization, the approaches still produce good solutions with a mild loss of at most 3% of average solution quality.

6.3.6. Comparison with Solutions from Practice. For reasons of confidentiality, we could not obtain reference solutions or current network costs for the instances presented above. Instead, a direct comparison with an instance of a European cross-docking network from a recent project has been conducted in cooperation with 4flow AG. The base network consists of 228 consumers, 545 suppliers, five hubs, and 5,857 edges, resulting in a tariff-expanded network with 209,304 edges. It is fully connected in contrast to the layered structure observed so far. In this instance, the AMIP-H framework obtained a solution with 1.2% gap to optimality. We compared this against a solution obtained with a standard software for supply chain design at project start operating on a conventional model. Our solution constitutes a 14% improvement, which, if applied on an annual basis, results in a savings up to 1.6 million euros.

7. Summary and Conclusions

The tactical transportation planning model presented in this paper integrates the important aspects of tactical logistics network optimization: realistic transportation tariffs, delivery patterns, and inventory costs. Several algorithmic techniques have been devised to address the challenges associated with the specific instance structure induced by our model. These methods have been successfully tested on a broad set of real-world instances.

Among our techniques, we propose a local search procedure that simultaneously reroutes flow of multiple commodities. Equipping the local search with different types of initial solutions, such as multicommodity flow patterns derived from a strengthened LP relaxation or from purely combinatorial path-based approaches, yields solutions that are within a single-digit percent of the optimum on average. Our algorithm can be used either in connection with standard MIP solvers for optimal solution quality or as a purely combinatorial algorithm, yielding competitive solutions without usage of third-party software. Hence, the broad spectrum of our algorithms offers a flexible tradeoff among solution quality, operating cost, and computation time.

The performance of our algorithms to a great part relies on the successful isolation of the tariff selection subproblem. We devise a variety of exact and heuristic methods to efficiently solve this problem, providing a tradeoff between speed and exactness of the solution procedure. A computational analysis of these algorithms and additional techniques can be found in a companion paper to this article by König, Matuschke, and Richter (2012) that also provides further theoretical insights into the tariff selection subproblem.

Currently, our algorithmic toolkit is being integrated into real-world software by our partner 4flow AG, and a second project that incorporates robustness aspects into the model has started.

Supplemental Material

Supplemental material to this paper is available at http://dx.doi.org/10.1287/trsc.2014.0541.

Acknowledgments

The authors would like to thank Cristina Hayden and Lars Stolletz of 4flow AG for fruitful discussions on the subject of logistics planning, providing data for our computational study, and offering their opinion when interpreting the solutions computed. The authors also thank Michael Bastubbe and Hendrik Lüthen for their help in implementing the algorithmic framework. This work was supported by the European Regional Development Fund and is part of a joint project of Technische Universität Berlin and 4flow AG.

References

Afentakis P, Gavish B (1986) Optimal lot-sizing algorithms for complex product structures. *Oper. Res.* 34(2):237–249.

Afentakis P, Gavish B, Karmarkar U (1984) Computationally efficient optimal solutions to the lot-sizing problem in multistage assembly systems. *Management Sci.* 30(2):222–239.

Blumenfeld DE, Burns LD, Daganzo CF, Frick MC, Hall RW (1987) Reducing logistics costs at General Motors. *Interfaces* 17(1):26–47.

Burns LD, Hall RW, Blumenfeld DE, Daganzo CF (1985) Distribution strategies that minimize transportation and inventory costs. *Oper. Res.* 33(3):469–490.

Cakir O (2009) Benders decomposition applied to multi-commodity, multi-mode distribution planning. *Expert Systems Appl.* 36(4): 8212–8217.

Çetinkaya S (2005) Coordination of inventory and shipment consolidation decisions: A review of premises, models, and justification. Pardalos PM, Hearn D, Geunes J, Akçali E, Romeijn HE, Shen Z-JM, eds. *Applications of Supply Chain Management and E-Commerce Research, Applied Optimization*, Vol. 92 (Springer, New York), 3–51.

Chakrabarty D, Chekuri C, Khanna S, Korula N (2011) Approximability of capacitated network design. Günlük O, Woeginger GJ, eds. *Integer Programming and Combinatoral Optimization* (Springer, Berlin Heidelberg), 78–91.

Chan LMA, Muriel A, Shen Z-JM, Simchi-Levi D, Teo C-P (2002) Effective zero-inventory-ordering policies for the single-warehouse multiretailer problem with piecewise linear cost structures. *Management Sci.* 48(11):1446–1460.

Chopra S, Meindl P (2007) *Supply Chain Management: Strategy, Planning, and Operations* (Pearson Prentice-Hall, Upper Saddle River, NJ).

Chouman M, Crainic TG, Gendron B (2011) Commodity representations and cutset-based inequalities for multicommodity capacitated fixed-charge network design. Technical report, CIRRELT-2011-56. Université de Montréal. Centre de recherche sur les transports, Montréal.

Clark AJ, Scarf H (1960) Optimal policies for a multi-echelon inventory problem. *Management Sci.* 6(4):475–490.

Costa AM (2005) A survey on Benders decomposition applied to fixed-charge network design problems. *Comput. Oper. Res.* 32(6):1429–1450.

Costa AM, Cordeau J-F, Gendron B (2009) Benders, metric and cutset inequalities for multicommodity capacitated network design. *Comput. Optim. Appl.* 42(3):371–392.

Crainic TG (2000) Service network design in freight transportation. *Eur. J. Oper. Res.* 122(2):272–288.

Crainic TG, Gendreau M (2002) Cooperative parallel tabu search for capacitated network design. *J. Heuristics* 8(6):601–627.

Crainic TG, Gendreau M, Farvolden JM (2000) A simplex-based tabu search method for capacitated network design. *INFORMS J. Comput.* 12(3):223–236.

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

Dobson G (1982) Worst-case analysis of greedy heuristics for integer programming with nonnegative data. *Math. Oper. Res.* 7(4): 515–531.

Fischetti M, Salvagnin D, Zanette A (2010) A note on the selection of Benders' cuts. *Math. Program.* 124(1–2):175–182.

Frangioni A, Gendron B (2009) 0-1 Reformulations of the multicommodity capacitated network design problem. *Discrete Appl. Math.* 157(6):1229–1241.

Geoffrion AM, Graves GW (1974) Multicommodity distribution system design by Benders decomposition. *Management Sci.* 20(5):822–844.

Geunes J, Pardalos PM (2003) Network optimization in supply chain management and financial engineering: An annotated bibliography. *Networks* 42(2):66–84.

Ghamlouche I, Crainic TG, Gendreau M (2003) Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design. *Oper. Res.* 51(4):655–667.

Ghamlouche I, Crainic TG, Gendreau M (2004) Path relinking, cycle-based neighbourhoods and capacitated multicommodity network design. *Ann. Oper. Res.* 131(1):109–133.

Guisewite G, Pardalos P (1990) Minimum concave-cost network flow problems: Applications, complexity, and algorithms. *Ann. Oper. Res.* 25(1):75–99.

Jans R, Degraeve Z (2007) Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches. *Eur. J. Oper. Res.* 177(3):1855–1875.

Jayaraman V (1998) Transportation, facility location and inventory issues in distribution network design: An investigation. *Internat. J. Oper. Production Management* 18(5):471–494.

Kempkes JP, Koberstein A, Suhl L (2010) A resource based mixed integer modelling approach for integrated operational logistics planning. Aalst W, Mylopoulos J, Rosemann M, Shaw MJ, Szyperski C, Dangelmaier W, Blecken A, Delius R, Klöpfer S, eds. *Advanced Manufacturing and Sustainable Logistics*, Lecture Notes in Business Information Processing, Vol. 46 (Springer, Berlin Heidelberg), 281–294.

Kim D, Pardalos PM (1999) A solution approach to the fixed charge network flow problem using a dynamic slope scaling procedure. *Oper. Res. Lett.* 24(4):195–203.

Kliewer G, Timajev L (2005) Relax-and-cut for capacitated network design. Brodal GS, Leonardi S, eds. *Algorithms–ESA 2005*, Lecture Notes in Computer Science, Vol. 3369 (Springer, Berlin Heidelberg), 47–58.

König FG, Matuschke J, Richter A (2012) Multi-dimensional commodity covering for tariff selection in transportation. Delling D, Liberti L, eds. *12th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2012), OASICS*, Vol. 25 (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany), 58–70.

Lueker GS (1975) Two NP-complete problems in nonnegative integer programming. Technical report, Princeton University Computer Science Laboratory, Princeton, NJ.

Magnanti TL, Wong RT (1984) Network design and transportation planning: Models and algorithms. *Transportation Sci.* 18(1): 1–55.

Richey MB, Parker RG (1986) On multiple Steiner subgraph problems. *Networks* 16(4):423–438.

Schöneberg T, Koberstein A, Suhl L (2010) An optimization model for automated selection of economic and ecologic delivery profiles in area forwarding based inbound logistics networks. *Flexible Services Manufacturing J.* 22(3–4):214–235.

Simchi-Levi D, Kaminsky P, Simchi-Levi E (2003) *Designing and Managing the Supply Chain: Concepts, Strategies, and Case Studies* (McGraw Hill, New York).

Stadtler H (2003) Multilevel lot sizing with setup times and multiple constrained resources: Internally rolling schedules with lot-sizing windows. *Oper. Res.* 51(3):487–502.

Wagner HM, Whitin TM (1958) Dynamic version of the economic lot size model. *Management Sci.* 5(1):89–96.