An Integrated Approach to Tactical Logistics Network Optimization*

Preprint 034-2012, TU Berlin, Institut für Mathematik

Tobias Harks† Felix G. König‡ Jannik Matuschke§ Alexander Richter§

Jens Schulz§

November 8, 2012

Abstract

We propose a new mathematical model for the optimization of logistics networks on the tactical level. Main features include accurately modelled tariff structures and the integration of spatial and temporal consolidation effects via a cyclic pattern expansion. By 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 re-routes 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 industry, we prove that most of our solutions are within a single-digit percentage of the optimum.

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 out 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 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). As this

*This work was supported by the European Regional Development Fund (ERDF) and is part of a joint project with 4flow AG, Berlin, Germany.

†Maastricht University, School of Business and Economics, PO Box 616, 6200 MD, The Netherlands, e-mail: `t.harks@maastrichtuniversity.nl`

‡TomTom International BV, An den Treptowers 1, 12435 Berlin, Germany, e-mail: `felix.koenig@tomtom.com`

§Technische Universität Berlin, Institut für Mathematik, Straße des 17. Juni 136, 10623 Berlin, Germany. e-mail: {`matuschke,arichter,jschulz`}`@math.tu-berlin.de`.

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. Logistics in particular occupy a central place in SCM, as transportation and storage of physical goods accounts for a significant share of the operational cost in a supply network. Moreover, recent trends to outsource logistic operations to international logistics providers and/or employ consultants to streamline logistics across the supply chain emphasize the need for a network-wide view.

Due to a 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 et al. 2003). The work presented in this paper is concerned with logistics planning on the *tactical* level. Here, it is commonly assumed that the supply chain is already in place: location and product decision have been made, and the general design of the supply chain network is fixed. Typical logistic 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 logistics planning in an integrated fashion, explicitly including realistic transport tariffs and the trade-off between inventory cost and economies of scale in transportation.

1.1 Problem Description

We proceed to give a general description of the task we refer to as tactical logistics network optimization 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 the cost incurred for a given amount of a commodity (mix). E.g., a full truck load 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, like 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 in order to utilize capacity efficiently and enable economies of scale (Çetinkaya 2005). Consolidation may occur over space as well as over 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 in order for more flow to arrive, thereby enabling a larger outbound shipment.

Since holding inventory also incurs cost, however, 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, like "once every month", "once every week" or "once every day of the week", and in each slot of the pattern (like in one month, week or weekday), deliveries are dispatched, and inventories are replenished or depleted.

All in all, the outcome of tactical logistics network optimization 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, and finally
- 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 which 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 Section 2, we propose a new model for the optimization of logistics 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. While 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. By 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 the next Section 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.

While the resulting network design problem can be naturally formulated as a MIP, 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: In order to identify cost efficient paths, our algorithms need good and fast estimates on the cost incurred by 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 Section 3, we will propose different algorithms that provide an efficient balance of accuracy and speed for solving this *NP*-hard subproblem.

In Section 4, we then 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 flow of a single commodity only), our approach applies a neighborhood search based on path decomposition of flow and re-routing multiple commodities simultaneously. In order 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, being able to generate solutions of reasonable quality for even the largest instances in short time. The second is more accurate in cost estimation and is therefore used as the central subroutine in our local search improving moves. By forbidding certain paths (for instance direct connections) and linearizing costs we further tune the initial solutions towards a high level of flow consolidation that will eventually be disaggregated by the local search heuristic.

In Section 5, we complement our heuristic approach by mixed integer programming techniques. As the plain MIP formulation is not suited for solving reasonably sized real-world instances due to 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 post-processing step to improve solution quality. Combining the 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 Section 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 industry 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 modelling and algorithmic techniques yield a cost reduction of over ten percent over the current status quo, which could result in annual savings of several millions of euros.

1.3 Related Work

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

Supply Chain Management

Literature on SCM is as broad and diverse as the field itself, see the textbooks by Simchi-Levi et al. (2003) and Chopra and Meindl (2007). Hence, we confine our literature review to works explicitly dealing with logistic issues that apply mathematical optimization techniques. 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 by Geoffrion and Graves (1974), the authors model a multi-commodity network with several plants, possible distributions center locations, and demand zones on the strategic level. Fixed costs are associated with opening distribution centers, plants are capacitated for each commodity, and there are upper and lower limits on the throughput of a distribution center. Transportation and production costs are commodity-dependent and assumed to be linear. The resulting mixed integer programming (MIP) model is solved using Benders decomposition.

An obvious drawback of the above model is the assumption that all transportation costs are linear. Bookbinder and Reece (1988) extend the work of Geoffrion and Graves (1974) by modeling transport from the distribution centers to the demand zones as capacitated vehicle routing problems. They use an iterative approach to solve the problem, where transportation costs to the demand zones is repeatedly estimated by solving a series of vehicle routing problems.

One important aspect missing from both above models is the possibility to hold inventory at some nodes of the network, which is essential to model temporal consolidation effects: Delaying transports from a facility until some inventory has accumulated may realize economies of scale effects, thereby reducing transportation cost.

Models with Inventory

One of the earliest strategic optimization models incorporating the interdependence of location, transportation, and inventory decisions in SCM 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. The model is solved using standard MIP solvers. Note that this still only provides a coarse approximation to economies of scale in transportation, as theoretically, also transportation modes with low delivery frequency could carry low shipping volume, making their assumed low per-unit cost unrealistic.

Kempkes et al. (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, costs can be piecewise constant or piecewise linear and may depend on various properties of the flow, like the mass and volume of material. Planning occurs over multiple however non-cyclic periods, and in particular, inventory cost is taken into account. The authors propose a flow-based construction heuristics to generate a first initial feasible solution that is passed to a standard MIP solver. In order to introduce all details necessary for realistic operational planning, their model even allows for logical relations between different resources, which however 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 one hundred commodities.

While most of the network-wide SCM models discussed earlier are focused on strategic planning and incorporate location decisions while modeling transportation and inventory decisions rather coarsely, 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, while 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 cost is minimized. Clearly, the fixed order cost in this model can be interpreted as the cost of one sufficiently large vehicle delivering goods in this stage. As shown by Wagner and Whitin (1958), this problem can be solved to optimality in polynomial time by dynamic programming. This basic model has been extended in many ways since then, and most variants are computationally hard, see e.g., Jans and Degraeve (2007) for an overview. The practical importance of considering the trade-off between transportation and inventory cost is highlighted impressively by Burns et al. (1985), Blumenfeld et al. (1987), where the authors were able to reduce logistics cost by 26% in a case study for General Motors.

Anily and Federgruen (1990) integrate the routing of vehicles with fixed capacity serving several customers along their route with inventory considerations. Chan and Simchi-Levi (1998) extended this setting to a three-stage distribution system with vehicle routing problems solved in between all stages. These works particularly emphasize the importance of detailed modeling in order to assess transportation costs more realistically when making inventory decisions.

Generalizing lot-sizing to networks with multiple stages brings it closer to the requirements of tactical planning in SCM. The first such model was introduced by Clark and Scarf (1960) and further developed by Afentakis et al. (1984), Afentakis and Gavish (1986). An overview of more recent works can be found by Stadtler (2003). Another trend in lot-sizing literature, reflecting some needs from tactical SCM, is modeling the availability of different transportation modes with different capacities and cost structures between a pair of nodes as considered by Jaruphongsa et al. (2005, 2007).

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 nodes is fixed a priori in all lot-sizing models, so the possibility for more spatial consolidation at hubs is effectively ignored. Hence, a significant gap to network-wide tactical logistics optimization remains.

Capacitated Network Design

While network flow seems to be the dominant aspect in logistics network optimization, the fixed cost nature of transportation brings in network design decisions: We have to install sufficient capacity in the network such that all flow can be routed. In 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 telecommunication 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 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 minimum edge cost flow problem (listed as [ND32] by Garey and Johnson (1979)), which does not even permit an approximation better than to a factor of $\Omega(\log \log n)$ in the number of nodes n (Chuzhoy et al. 2008). 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 logistic 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 et al. (2000) proposed a tabu search procedure based on a neighborhood in the multi-commodity flow polytope. Their algorithm has later been adapted for parallelization by Crainic and Gendreau (2002). A different neighborhood for tabu search was introduced by Ghamlouche et al. (2003), operating on the network design and modifying it along cycles. This procedure has been refined by the same authors by supplementing it with a path relinking technique (Ghamlouche et al. 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 et al. (2004) generalize this technique to multi commodity 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 et al. (2000).

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 merits of strong linking inequalities. Chouman et al. (2009) present lifting procedures for *strong capacity* and *network cutset inequalities* for a fixed charge network flow problem in which the edge capacities are one-dimensional. Another promising technique to solve multi-commodity network flow problems is to apply a Benders decomposition, see for instance Costa (2005), Cakir (2009). For a multi-commodity capacitated network flow problem Costa et al. (2009) show the relation between different classes of inequalities, especially how the inequalities from (non-extreme) 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 et al. (2008) and Fischetti et al. (2010) propose in case of an infeasible subproblem to find a minimal infeasible subsystem, which is an NP-hard problem. They show that this idea can be integrated in 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 cost on the edges. However, we extend the standard concepts of capacity and cost to more generality in order to reflect the requirements of logistics modelling more precisely. Moreover, we expand the underlying network significantly in order to model delivery patterns, inventory effects, and complex transport tariffs. We proceed to detail all of these features in the following sections. Finally, we discuss the advantages and inherent challenges of our model.

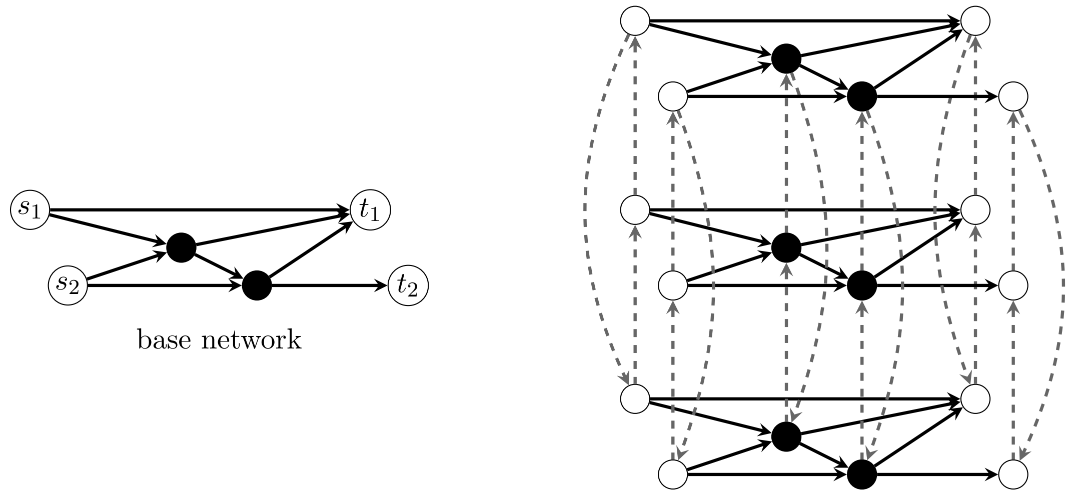
2.1 Pattern Expansion

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 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 which will take place again and again over time, a solution should be a *cyclic* pattern for dispatching deliveries and replenishing and depleting inventories. To integrate temporal and spatial consolidation together 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 a 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 conceptually not differentiate between holdover edges and transport edges. We denote the set of nodes in the pattern expanded network by $\mathcal{V}$ and the set of all edges of $\mathcal{G}$ (also called transport relations) by $\mathcal{T}$.

We illustrate this cyclic construction with an example. Consider the base network in Fig. 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

Figure 1: Network Expansion



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

additional holdover edges as illustrated in Fig. 1 (b). Note that the pattern expansion need not be symmetric in the sense that all holdover edges between any two adjacent copies of a node exist. Our model includes the possibility of deleting some holdover edges or even some transport edges of the base network in a time slot. In the example, deleting a holdover edge from B_3 to B_1 may be necessary if it is not allowed to store commodities at the corresponding node over the weekend.

Low transportation costs may be achieved by cleverly consolidating shipments at intermediate nodes in different time slots. This may, on the other hand come with high inventory costs as some shipments might have to be delayed or productions have to be made ready earlier for consolidation. For efficient spatial and temporal consolidation the difficulty is to strike the right balance between transportation costs and inventory cost.

2.2 Commodities and Properties

Commodities in a logistics network can be very diverse, e.g., in their size, weight, or value, and logistic 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 the next section 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. Note, that together with the concept of containers introduced in the following section the flexibility of properties can also be used to model more abstract aspects of a commodity, such as "needs cooling", "is hazardous", or the like by introducing a corresponding property. Such commodities receive a strictly positive extent in this property and adjusting container capacity w.r.t. this property can be used to allow or restrict the loading of these commodities.

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 is expressed by $\alpha(x) \in \mathbb{R}_+^P, \alpha_j(x) := \sum_{i \in K} \alpha_{ij} x_i$.

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 $v \in \mathcal{V}$ (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. We identify each transport relation $T \in \mathcal{T}$ with the set of its *tariffs* available for transporting a flow of commodities from start(T) to end(T). Each such tariff $t \in T$ corresponds to 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.,

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

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 $\mathcal{F}_b$ be the set of all multicommodity flows in the pattern expanded network satisfying node balances b. Then our goal is to find an optimal solution to

$$\begin{array}{llcll}
\min & \displaystyle\sum_{T \in \mathcal{T}} \sum_{t \in T} C_t(x(t)) & & & \\
\text{s.t.} & (x(T))_{T \in \mathcal{T}} & \in & \mathcal{F}_b & \\
& \displaystyle\sum_{t \in T} x_i(t) & = & x_i(T) & \forall t \in T, \forall i \in K \\
& x(t) & \geq & 0 & \forall t \in T, T \in \mathcal{T}
\end{array}$$

where $x(T) \in \mathbb{R}_+^K$ is the total flow on each transport relation $T \in \mathcal{T}$ and $x(t) \in \mathbb{R}_+^K$ is the amount of flow transported using tariff $t \in T$.

We will now present a set of cost functions that covers most tariffs occurring in today's logistical applications. In the next section, we will then show how all 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 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

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

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 full truck load carriers (FTL) 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

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

In practice, logistic carriers offer groups of such tariffs realizing different levels of discount for higher shipment volumes. We will see in Section 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 which incurs 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.

Often, per-unit shipping rates decrease with increasing size of the shipment. For linear cost functions, this can manifest in either *incremental discounts* or *all-unit discounts.*

Incremental discount costs. We consider a tariff with varying cost rates depending on a single property, which are specified on intervals. The cost rates are decreasing with increasing size of shipment, resulting in a piecewise linear and concave cost function. Formally, let L be the set of levels, such that c^ℓ is the cost rate on the interval $[\beta_j^\ell, \beta_j^{\ell+1}]$ for the fixed property $j \in P$. Then the cost function is

$$C(x) = \sum_{\ell \in L} c^\ell \cdot \min \left\{ \beta_j^{\ell+1} - \beta_j^\ell,\ \max \left\{ \alpha_j(x) - \beta_j^\ell, 0 \right\} \right\}.$$

All-unit discount costs. Again we consider linear cost rates in some property j 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 increasing the transport volume to the beginning of the next level reduces overall transportation cost (this corresponds to the common practice of declaring higher volumes than actually transported in such cases (Chan et al. 2002); also cf. Table 1 for a graphical illustration of the resulting cost function). Formally, if cost rate c^ℓ for $\ell \in L$ is applicable starting from transport volume β_j^ℓ on, the cost function is

$$C(x) = \min_{\ell \in L} \left(c^\ell \cdot \max \left\{ \alpha_j(x), \beta_j^\ell \right\} \right) .$$

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 (MIP) can easily be obtained from this description, making it accessible to MIP based solving techniques, while its compact structure effectively demonstrates the degree of mathematical uniformity achieved in modelling.

We will 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 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, and let $\beta_j(e)$ be the corresponding capacity of a container at edge e. Then the *capacity constraints*

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

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 tarriff expansion to zero for each commodity. We thus obtain the *flow conservation constraints*

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

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 a MIP.

$$\begin{array}{llcll}
\min & \displaystyle\sum_{i \in K}\sum_{e \in E} c_i(e)x_i(e) + \sum_{e \in E} c(e)y(e) & & & \\
\text{s.t.} & \displaystyle\sum_{e \in \delta^+(v)} x_i(e) - \sum_{e \in \delta^-(v)} x_i(e) & = & b_i(v) & \forall v \in V, i \in K \\
& \displaystyle\sum_{i \in K} \alpha_{ij} x_i(e) & \leq & \beta_j(e)y(e) & \forall e \in E, j \in P \\
& y(e) & \leq & u(e) & \forall e \in E \\
& x_i(e) \in \mathbb{R}_+,\ y(e) \in \mathbb{Z}_+ & & & \forall e \in E, i \in K
\end{array}$$

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 $\text{start}(T)$ to $\text{end}(T)$ 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 Modelling Tariffs with Containers

We now proceed to explain how containers can be used to accurately model the different types of transport tariffs introduced in the previous section.

It is clear that both commoditiy-dependent linear costs and property-dependent piecewise constant costs are directly captured by the container concept. Linear costs are part of the definition, while 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.

In order to model the maximum over multiple cost rates we need to introduce a *fractional containers*, i.e., for this container the variable $y(e)$ corresponding to the number of installed copies can be fractional. When setting 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 gadget requires $y(e)$ to be set to $\max_{j \in P} \alpha_j(x_i(e))/\beta_j(e)$, which is equal to the cost function by choice of $\beta_j(e)$. Introducing such fractional containers does not have significant impact on the complexity of the model. However, since the instances in our computational study do not make use of such tariffs, for the sake of simplicity we will assume throughout the paper that all containers have to be installed in integral increments.

Piecewise linear concave functions arising from incremental discount tariffs can be interpreted as the minimum of several functions with both linear and fixed cost. Defining

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

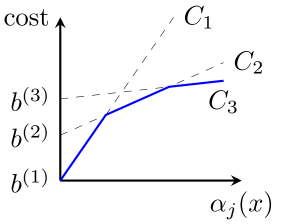
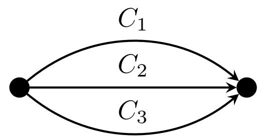
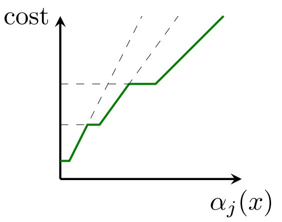
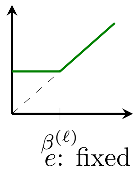
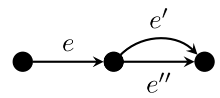
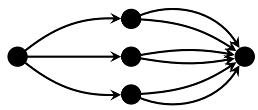
tariff	cost illustration	gadget
incremental discount (piecewise linear concave)	 $C = \min\{C_1, C_2, C_3\}$	minimum through parallel edges  $C_\ell(x) := c^{(\ell)}\alpha_j(x) + b^{(\ell)}$
all-unit discount		subgadget for $c^{(\ell)} \cdot \max\left\{\alpha_j(x),\ \beta^{(\ell)}\right\}$   e: fixed cost, infinite capacity e': linear cost, infinite capacity e'': no cost, capacity $\beta^{(\ell)}$ minimum through parallel gadgets 

Table 1: Modelling complex transport tariffs with containers.

it is easy to verify that $C(x) = \min_{\ell \in L} C_\ell(x)$ (cf. Table 1 for an illustration). We now introduce a gadget of $|L|$ parallel edges $e_1, \dots, e_{|L|}$ with $c(e_\ell) = b^{(\ell)}$, $c_i(e_\ell) = \alpha_{ij} c^{(\ell)}$ and $\beta_j(e_\ell) = \beta_j^{(\ell+1)}$ (all other capacities are left infinite). Now 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.

Note that functions of the form $c^{(\ell)} \cdot \max\left\{\alpha_j(x),\ \beta^{(\ell)}\right\}$ can be modeled by the following gadget (see the subfigure in Table 1): 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 $c(e) = c^{(\ell)}\beta^{(\ell)}$ and $c(e') = c(e'') = 0$. We also set the linear costs $c_i(e) = c_i(e'') = 0$ and $c_i(e') = \alpha_{ij} c^{(\ell)}$ for all $i \in K$. Capacity $\beta_j(e'')$ is set to $\beta^{(\ell)}$, all other capacities are left infinite. Now, all-unit discount tariffs, which can be represented as minimum of such functions, can be easily modelled by introducing several of these gadgets in parallel.

Table 1 gives an overview of the tariffs which can be represented in our model.

3 Tariff Selection Subproblem

While containers constitute a versatile tool to model various transport tariffs as described in Section 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 due to tariff gadgets by encapsulating tariff selection decisions in a subordinate optimization problem, that we call the *tariff selection* subproblem (TS). While some of our algorithms for TTP introduced in Sections 4 and 5 will operate directly on tariff gadgets as introduced in Section 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 $T \in \mathcal{T}$: Given a fixed vector $\bar{x}(T) \in \mathbb{R}_+^K$ of flow to be transported on T, it asks which transport tariffs should be selected and how should the fixed demand be distributed among selected tariffs in order to meet flow demand at minimum cost? More formally, the problem TS for transport relation $T \in \mathcal{T}$ can be stated as

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

Note again that we slightly abuse notation to denote by T both a transport relation and the set of available tariffs on it. A solution to TS comprises a vector $x(t) \in \mathbb{R}_+^K$ of multi-commodity flow for each tariff $t \in T$ such that their sum meets the total flow demand $\bar{x}(T)$. In a network perspective, solving the union of the TS problems on all transport relations optimizes transport cost with respect to a given fixed multi-commodity flow in the pattern expanded network.

Depending on which of the five types of tariff cost functions introduced in Section 2.3 are present in TS, we employ different techniques in order to solve TS. In Section 3.1 we devise a mixed integer programming formulation for arbitrary combinations of tariff cost functions in TS. However, out of the different tariff cost functions, property-dependent piecewise constant cost stands out for a number of reasons. First, while 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 (cf. Proposition 3.1). Second, it may be the tariff type occurring most frequently in logistic applications: Indeed, in the real-life data for our computational study in Section 6, many transport relations are equipped exclusively with piecewise constant tariffs. Therefore, Section 3.2 is devoted to theoretic and algorithmic insights into TS for this tariff type.

Elaborate algorithms for TTP, which we present later in Section 4, solve TS as a subroutine very frequently. Due to 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 Section 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 Section 3.2.2.

3.1 MIP for the General Case

The introduction of tariff gadgets in Section 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 $t \in T$ is represented by a container gadget $(V(t), E(t))$, as detailed in Section 2.3, we denote with $E(T) := \bigcup_{t \in T} E(t)$ respectively $V(T) := \bigcup_{t \in T} V(t)$ the set of all edges respectively nodes that are introduced to model the tariff structure on transport relation T. TS can then be written as

$$
\begin{aligned}
\min \quad & \sum_{e \in E(T)} c(e)y(e) + \sum_{i \in K} c_i(e)x_i(e) && \\
& \sum_{\substack{e \in E(T) \cap \\ \delta^+(\text{start}(T))}} x_i(e) = \bar{x}_i(T) && \forall i \in K \\
& \sum_{\substack{e \in E(T) \cap \\ \delta^-(\text{end}(T))}} x_i(e) = \bar{x}_i(T) && \forall i \in K \\
\sum_{e \in \delta^-(v)} x_i(e) - & \sum_{e \in \delta^+(v)} x_i(e) = 0 && \forall v \in V(T) \setminus \{\text{start}(T), \text{end}(T)\}, \quad i \in K \\
& \sum_{i \in K} \alpha_{ij} x_i(e) \leq \beta_j(e) y(e) && \forall e \in E(T), \quad j \in P \\
& y(e) \leq u(e) && \forall e \in E(T) \\
& y(e) \in \mathbb{Z}_+, \; x_i(e) \in \mathbb{R}_+. &&
\end{aligned}
$$

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

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

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

It is not hard to see that solving TS to optimality remains *NP*-hard here, even for very restricted special cases. We give a straight-forward reduction from the well-known unbounded knapsack problem, which is proven to be *NP*-hard (Lueker 1975), to TS instances with only a single property and a single commodity.

Proposition 3.1. *Problem TS is NP-hard, even when restricted to instances with only piecewise constant cost functions, a single property and a single commodity.*

Proof. Proof. In the single commodity, single property case the assignment variables $x(e)$ for the commodity are not relevant, since a feasible assignment can trivially be found as long as installed container capacities cover the amount of properties resulting from the commodity demand. The remaining difficulty is find an optimal combination of container copies. We use this insight to reduce the unbounded knapsack problem to this special TS by introducing knapsack items for every container type occurring in TS.

In an instance I of the unbounded knapsack problem, one is given n item types with integer values $v_1, \ldots, v_n$ and weights $w_1, \ldots, w_n$, and a weight capacity W. The goal is to select an positive integral number z_e for each item type e, such that the total value of packed items $\sum_{e=1}^{n} v_e z_e$ is maximized without violating the capacity constraint $\sum_{e=1}^{n} w_e z_e \leq W$. In the decision version of the problem, the question is whether feasible values z_e exist such that $\sum_{e=1}^{n} v_e z_e \geq V$ for a desired value $V \in \mathbb{R}_+$.

Given I, we construct an instance J of TS of the above special case as follows: first, define $u_e := \lceil W/w_e \rceil$ to be the maximum number any item can occur in a feasible knapsack solution. Then, for each item $e \in \{1, \ldots, n\}$, we introduce a tariff modeled by one container e with fixed cost $c(e) := v_e$ and capacity $\beta(e) := w_e$. Moreover, we set $\bar{x}(T) := \sum_{e=1}^{n} w_e u_e - W$ and $\alpha := 1$.

We now argue that I possesses a solution with value at least V, if and only if J can be solved with cost at most $\sum_{e=1}^{n} v_e u_e - V$. The main idea is that we think of instance J as an unpacking problem. Imagine all items have been selected u_e times and we choose y_e items of type e to be unpacked to fulfill the unpacking demand of $\bar{x}(T)$, which means that we are allowed to leave at most W of the total weight $\sum_{e=1}^{n} w_e u_e$ packed. First assume there is a feasible solution z to I with value at least V. We define $y(e) := u_e - z_e$ and $x(e) := w_e y(e)$ for each $e = 1, \ldots n$. Now (x, y) yields a feasible solution for J since

$$\sum_{e \in E(T)} x(e) = \sum_{e=1}^{n} w_e y(e) = \sum_{e=1}^{n} w_e (u_e - z_e) \geq \sum_{e=1}^{n} w_e u_e - W = \bar{x}(T)$$

$$\text{and} \quad \alpha x(e) = 1 \cdot w_e y(e) = \beta(e) y(e)$$

Note that any excess flow in a container e can feasibly be removed from an optimal solution without changing cost. Moreover, the cost for all selected containers is

$$\sum_{e \in E(T)} c(e) y(e) = \sum_{e=1}^{n} v_e (u_e - z_e) = \sum_{e=1}^{n} v_e u_e - V.$$

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 in order to "cover" the capacity extents of a fixed flow vector $\bar{x}(T)$—gives rise to consider our algorithms a generalization of the natural greedy approach to integer programs with nonnegative data as studied for example by Dobson (1982).

The greedy algorithm for tariff selection repeatedly selects a "most efficient" container $e \in E(T)$ to cover portions of, or the whole remaining commodity demand $\bar{d}$, initialized by $\bar{d} := \bar{x}(T)$. Here, "efficiency" of a container is measured by the function `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 `Fill(`e,$\bar{d}$`)`, which returns a mix of commodities $\Delta \in \mathbb{R}_+^K$, with $\Delta_i \leq \bar{d}_i$ for all $i \in K$ and $\alpha_j(\Delta) \leq \beta_j(e)$ for all $j \in P$, so as to ensure an "efficient" capacity usage 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., $\bar{d}$ is reduced to zero. While iterating, at some point in the algorithm there might be containers large enough to cover all remaining demand $\bar{d}$, while the `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: among all containers that suffice to cover $\bar{d}$, the algorithm picks the one with least cost and constructs a *complete solution* to be stored separately (if it is better than any previous solution), while it also proceeds with the partial solution arising from selecting the container with best `Score` value.

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

Algorithm 1: Greedy algorithm for tariff selection

Input: a TS subproblem on transport relation T with demand $\bar{x}(T)$

Output: assignment commodity vectors $x'(e) \in \mathbb{R}_+^K$, multiset Y' over $E(T)$

$\bar{d} \leftarrow \bar{x}(T)$; `// remaining uncovered demand`

$x(e) \leftarrow 0, \forall e \in E(T)$; $Y \leftarrow \emptyset$; `// current partial solution`

$x'(e) \leftarrow 0, \forall e \in E(T)$; $Y' \leftarrow \emptyset$; `// current best complete solution`

while *there is uncovered demand* $\bar{d}$ **do**

    if *there exists* $e_F = \operatorname{argmin}_{e \in E(T): \alpha(\bar{d}) \leq \beta(e)} c(e)$ **then** `// there is` e_F `that can store` $\bar{d}$

        if $Y' = \emptyset$ *or* $c(Y \cup e_F) < c(Y')$ **then** `// found new best solution?`

            replace Y' with $Y \cup e_F$ and $x'(e)$ with $x(e)$, $\forall e \in E(T)$;

            $x'(e_F) \leftarrow x'(e_F) + \bar{d}$; `// update new best solution`

    $e_B \leftarrow \operatorname{argmax}_{e \in E(T)}$`Score(`$e$,$\bar{d}$`)` ; `// pick most efficient container`

    $\Delta \leftarrow$ `Fill(`e_B , $\bar{d}$`)`; `// compute mix of commodities to assign`

    $\mathsf{n} \leftarrow \left\lfloor \min_{i \in K: \Delta_i \neq 0} \frac{\bar{d}_i}{\Delta_i} \right\rfloor$; `// 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` and thus 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

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

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

$$i = \operatorname{argmin}_{\,i'\in K}\|\bar{\beta}(e) - \min\{\texttt{linFrac(}\bar{d}^{i'},\ \bar{\beta}(e)\texttt{)},1\}\cdot\alpha(\bar{d}^{i'})\|\,,$$

where $\bar{d}^i$ is defined as $\bar{d}^i := (0,\dots,\bar{d}_i,\dots,0)$, and adds this amount of commodity i to the current vector Δ. Phase 1 might incur an unnecessary amount of slack in some capacities due to 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 $\ell \in K$, we define $\lambda_\ell := \langle\sum\Delta_i u_i, u_\ell\rangle/\|u_\ell\|^2$. Note that $\lambda_\ell u_\ell$ corresponds to the projection of $\sum\Delta_i u_i$ on $V(u_\ell)$. If $\lambda_\ell < 0$, we augment Δ by $\min\{-\lambda_\ell, \bar{d}_\ell\}$ 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, but merely 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 speed ups 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 $\bar{\alpha} := \alpha(\bar{x}(T))$ induced by the flow vector $\bar{x}(T)$. The result of this relaxation is the following covering problem (CR):

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

We can heuristically solve this problem very efficiently by adjusting Algorithm 1 to directly operate on the property vector $\bar{\alpha}$, that is 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 Combinatorial Heuristics

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 non-binary design-variables does not suffice to cope with the large instance sizes occuring 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 a cyclic expansion of network, yielding 250,000 edges on average per instance. This makes it very hard for heuristics that operate in the tariff expanded network without knowledge of the tariff structure. While most methods known from literature either work directly on the design variables or re-route 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 re-routes multiple commodities simultaneously.

In order to obtain good initial solutions for the local search algorithm that is presented in Section 4.3, we also provide two successive shortest path type algorithms, one that linearizes costs (`SPLC`) by estimating the per unit cost (Section 4.1) and one, denoted by `SPTS`, that uses a tariff selection method for this purpose (Section 4.2). The first method was designed with an emphasis on speed and low memory requirement, while the second is more accurate in cost estimation and is therefore 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 re-routing it separately. But 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. In order to address this issue, we adapt the two path-based algorithms to encourage consolidation by (i) forbidding the direct connections (which is well-suited in our types of practical networks) in the `SPLC` heuristic and (ii) using a partial linearization technique for the `SPTS` heuristic. Both refinements yield considerable improvements in solution quality of the local search procedure as we will see in Section 6.

4.1 Shortest Paths with Linearized Costs (SPLC)

A straightforward idea for obtaining shortest path edge weights is estimating the per unit shipping cost on each arc 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 minimal amount of time.

In each iteration, the algorithm chooses a commodity and finds a shortest path from a source to a sink. 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 be the commodity that is currently being routed and (x, y) be the current (partial) solution. For arc $e \in E$, we compute the *residual capacity for commodity* k:

$$\rho(e) = \min_j \frac{\beta_j(e)y(e) - \sum_i \alpha_{ij}x_i(e)}{\alpha_{kj}}.$$

If there is residual capacity $\rho(e) > 0$, we set the edge capacity $r(e) = \rho(e)$ and the edge weight $w(e) = c_k(e)$. If no residual capacity is left ($\rho(e) = 0$) another copy of this container can be selected if $y(e) < u(e)$. In this case we set

$$r(e) = \min_j \frac{\beta_j(e)}{\alpha_{kj}} \quad \text{and} \quad w(e) = c_k(e) + \frac{c(e)}{r(e)}.$$

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

Once a shortest path from a source to a sink of commodity k w.r.t. the weights w is found, a maximum amount of flow of commodity k w.r.t. the bottleneck of edge capacities r on this paths is sent. Note, that all computations above can be carried out very efficiently and of course, instead of updating weights and capacities of all edges in each step, these are calculated on-demand and only invalidated when neccessary.

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, this 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 Section 3.

Consolidation by forbidding direct connections (`SPLC-F`)

The `SPLC` heuristic favors large containers with low per unit cost rates and prefers direct connections as single detours cannot yield lower per unit cost. 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

Algorithm 2: Successive shortest path algorithm with linearized costs (SPLC)

Initialize $x = 0$, $y = 0$.
for *each commodity* $i \in K$ **do**
  Invalidate $r(e)$ and $w(e)$ for all $e \in E$.
  while *there is a source* s *of* i *with remaining supply* **do**
    Find path P in G from s to a sink t with $\sum_{e \in P} w(e)$ minimum.
    Augment x along P by $\min_{e \in P} r(e)$ units of commodity i, adjust y accordingly.
    Invalidate $r(e)$ and $w(e)$ for all $e \in P$.

detours can be easily identified and corrected by improving moves of the local search procedure.

4.2 Shortest Paths with Tariff Selection (SPTS)

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 already during the shortest path search. Although this more sophisticated approach requires more computational effort, it still turns out to be very 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

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

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, arc 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 Section 3, while 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.

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 $M := \min_j \frac{\sum_i \alpha_{ij} \Delta_i^+}{\sum_i \alpha_{ij} \Delta_i}$. For each available tariff t on a transport relation, we now compute the cost $C_t(\Delta^+)$ for routing Δ^+ and divide

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 G from s to t w.r.t. c̃, where c̃(T) is the estimated
    cost for augmenting the current flow x(T) by Δ(s,t) on transport relation T.
    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.
```

it by M to obtain an edge cost that anticipates future consolidation on this transport relation.

4.3 Paths-based Local Search

In the following we introduce a local search algorithm that re-routes flow along paths with the aim at improving feasible solutions. But before we describe the local search procedure let us formalize the concept of a paths based neighborhood and a path decomposition of network flow. 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 $T_1, \dots, T_m$ such that $\text{start}(T_{i+1}) = \text{end}(T_i)$ 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 $x(T) = \sum_{P \in \mathcal{P}: T \in P} \Delta_P$.

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 T of the path, $x(T)$ is decreased by Δ_P and the tariff selection of T is adapted accordingly, using the greedy tariff selection heuristic presented in Section 3. After removing a set of paths, the resulting partial solution is completed again by computing new paths using the SPTS heuristic introduced in Section 4.2. The move is accepted if the total cost of the solution decreases, and reverted otherwise.

We implemented two variants of improving moves:

Type A moves simply remove a single path at a time. This way, only small amounts of flow are re-routed 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 type A and B. This allows us to re-compute the path decomposition at the beginning of each phase, adapted to each movement type. In both cases paths are constructed in a DFS manner:

At a node in the DFS tree for each incident edge T we compute the maximal flow vector $\Delta(T)$ 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 $\Delta(T)$. 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, due to 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 are repeated alternatingly until the relative improvement achieved by both of them 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 re-routing the corresponding flow using a variation of type B moves.

5 MIP-Based Approaches

In this section, we discuss mixed integer programming 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 Section 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 (Section 5.1). We combine this with efficient preprocessing techniques to tighten the relaxation (Section 5.2). In Section 5.3 we use solutions to the LP relaxation of this strengthened aggregated formulation as initial solutions for our local search. Last, a post-processing step that improves solution quality is presented in Section 5.4. During this post-processing 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 multi-commodity capacitated network flow problems is to use a Benders Decomposition, see e.g., Costa et al. (2009). Preliminary runs with a Benders Decomposition combined with heuristics and adding additionally valid inequalties implemented in SCIP version 2.0 suffered from slow solving times. Interestingly, the subproblems (multi-commodity multi-dimensional 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 sized 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 that make the difference here. We leave this to future research on how to incorporate multi-dimensional capacities into combinatorial approaches similar to Costa et al. (2009).

5.1 Tariff Aggregated MIP (AMIP)

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 $e \in E(v, w)$ by 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

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

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 Section 3). Computational experiments on practice instances reveal that by applying a tariff selection heuristic on each relation, we can derive feasible solutions of the original model with only a minimal increase in cost. On the other hand, 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 to reduce problem sizes, the considered MIPs still suffer from numeric instability and weak lower bounds. We address these issues in the following paragraphs with two preprocessing steps that can be applied to the aggregated formulation.

Strengthened container inequalities

In order to strengthen the MIP formulation, valid inequalities can be added, see e.g., Chouman et al. (2009), among them are strong capacity inequalities and network cutset inequalities. We extended these inequalities to TTP. Using the demands b_i per commodity i then on every edge e the *strong capacity inequalities* must hold, i.e., $x_i(e) \leq b_i y(e)$. In our practical data with huge demands and given the frequency expanded networks, the demand values b_i are much larger than the ratio of capacity per property extend. These inequalities only yield a tighter formulation if the relation between α_{ij} and $\beta_j(e)$ is larger than the demand b_i, which is usually not the case in our instances.

Another set of valid inequalities is given by the *minimum-cardinality-inequalities* that express that enough egdes must be installed on every cut $S \subset V$ such that the demands from a source-sink-pair with source inside and sink outside the set S can be routed. As already observed by Chouman et al. (2009) these inequalities are weak if the magnitudes of the capacities vary widely as it is the case in our practical data. Their suggested improvements cannot be applied here as their model contains only binary $y(e)$-variables whereas ours are integer. In the following we show how we strengthen our capacity inequalities using similar ideas.

Solutions to the linear programming relaxation 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 close to zero 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 T an upper bound $\Gamma(T)$ on any feasible commodity flow $x(T)$ is known. Useful upper bounds Γ can be derived for transport relations incident to node sets $S \subset V$ that have either $\delta^+(S) = \emptyset$ or $\delta^-(S) = \emptyset$. An upper bound $\Gamma(T)$ provided, we can bound for every container $e \in E(T)$ its capacity $\beta_j(e)$ by $\beta_j(e) - s_j$ with s_j the result of solving

$$
\begin{array}{lll}
\min & s_j & \\
\text{s.t.} & \displaystyle\sum_{i \in K} \alpha_{ij'} x_i(e) + s_{j'} = \beta_{j'}(e) & \forall\, j' \in P \\
& 0 \leq x_i(e) \leq \Gamma_i & \forall\, i \in K \\
& s_{j'} \geq 0 & \forall\, j' \in P\,.
\end{array}
$$

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

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. In the aggregated formulation the capacity restricting inequalities involve many flow variables where property coefficients vary 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 as follows:

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

The scaled problem instance is equivalent to the non scaled 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 (ALP)

In Section 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 few sources or sinks only (such as direct source-sink connections). This implicitly encourages flow to take detours on non source-sink paths, where less strong 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 Section 6.

5.4 Pattern Optimization Subproblem

In the tariff selection subproblem considered in Section 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 $v_1, \dots, v_F$ be the copies of node v created in the pattern expansion step, with $v_i \in V(B_i)$ for $i \in \{1, \dots, F\}$. We consider the *Pattern Optimization Subproblem* induced by a fixed pair of nodes s, $t \in B$ and therefore define

$$V(s,t) := \bigcup_{i=1,\dots,F} \{s_i, t_i\}, \qquad \mathcal{T}(s,t) := \{T \in \mathcal{T} : \mathrm{start}(T), \mathrm{end}(T) \in V(s,t)\}.$$

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

$$\begin{array}{lrcll}
\min & \displaystyle\sum_{T \in \mathcal{T}(s,t)} \sum_{t \in T} C_t(x(t)) & & & \\
\text{s.t.} & (x(T)) \text{ and } (\bar{x}(T)) & & & \text{together are a feasible flow} \\
& \displaystyle\sum_{t \in T} x_i(t) & = & x_i(T) & \forall\, t \in T,\, T \in \mathcal{T}(s,t), \forall\, i \in K \\
& x(t) & \geq & 0 & \forall\, t \in T, \forall\, T \in \mathcal{T}(s,t).
\end{array}$$

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 in 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, non-disjoint pairs of nodes, as holdover edges of a common node appear in each of the problem as variables. Consequently, the order of the node pairs considered plays an important role.

We order the node pairs non-increasingly 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., $\sum_{j \in P} w_j \alpha_j (\sum_{T \in \mathcal{T}(s,t)} \bar{x}(T))$, 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 due to time constraints.

6 Computational Study

We verify the TTP model and the algorithmic approaches presented in the preceeding 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-sized 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, Retail). All base networks correspond to European supply chains in which goods are transported according to full truck load (FTL) or less than truck load (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 towards 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.

While 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 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.

set	#nodes in base network			#comm.	#edges		
(#instances)	#source	#sinks	#hubs		base	pattern exp.	tariff exp.
Auto1 (36)	35	6	7	162	335	2296	76653
Auto2 (18)	34	3	4	117	186	1364	29264
Chemical (50)	7	244	19	101	6601	41222	239238
Retail (41)	4	177	26	307	5665	35229	511064

Table 2: Average sizes of the instances per set

For future research, the instance library will be available upon 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 Sections 4 and 5 in order 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. In order to enable the evaluation of multiple scenarios, our industrial partner set a time limit of 30 min. 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 (cf. Section 6.2.1);

`MIP` Plain MIP formulation for comparison purposes, see Section 2;

`ALP` LP relaxation of aggregated MIP formulation (cf. Section 5.3)

`SPLC` Shortest path heuristic with linearized cost (cf. Section 4.1);

`SPLC-F` same as `SPLC`, but with forbidden direct connections (cf. Section 4.2);

`SPTS-L` Shortest path heuristic with tariff selection (cost estimator) and partial linearization (cf. Section 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 DualCore-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 the following Section 6.2.1 we elaborate on the interplay of the MIP and the local search heuristic, whereas the detailed settings for the variants of local search procedures are presented in Section 6.2.2.

6.2.1 Branch and Bound Frameworks.

Our tests involved different MIP formulations, which we implemented wthin CPLEX. Besides the plain MIP formulation (`MIP`) for comparison with our algorithms, we tested the aggregated MIP formulation including the preprocessing methods described in Section 5.2. Within this framework, we integrated our heuristics. The resulting algorithm is denoted by `AMIP-H`, and details of the implementation are given below. In order to obtain reasonably tight lower bounds, we also ran the aggregated MIP formulation without heuristic callbacks (`AMIP-B`). We invoked a time limit of 2 h for the branch and bound process, and an extra time of 1 h for applying local search and pattern optimization each.

When solving the aggregated and preprocessed MIP formulation from Section 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

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

Table 3: Average improvement of the lower bound compared to `MIP`. In parentheses, the number of instances handled by `MIP`/number of all instances per set is shown.

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 (cf. Section 5.1). We further improve this solution by applying the local search heuristic and pattern optimization with a time limit of 300 s.

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 Section 3.2.2 in order to evaluate the potential of a given LP solution to improve on the currently best solution: Only if the estimated total cost is within 8% to the best known solution, 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 Section 4.3 using initial solutions constructed by the aforementioned heuristics. The current tariff selection on the transport relations is then further improved using the exact MIP-formulation as described in Section 3. Finally, pattern-optimization is performed on the returned solution using the non-aggregated 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 min (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.

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.

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.

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)

Table 4: Average gaps to best known lower bound in %, number of achieved best solutions in parentheses.

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 towards 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 of up to 17% on average on set Auto1, while 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 Appendix.

Quality of Solutions

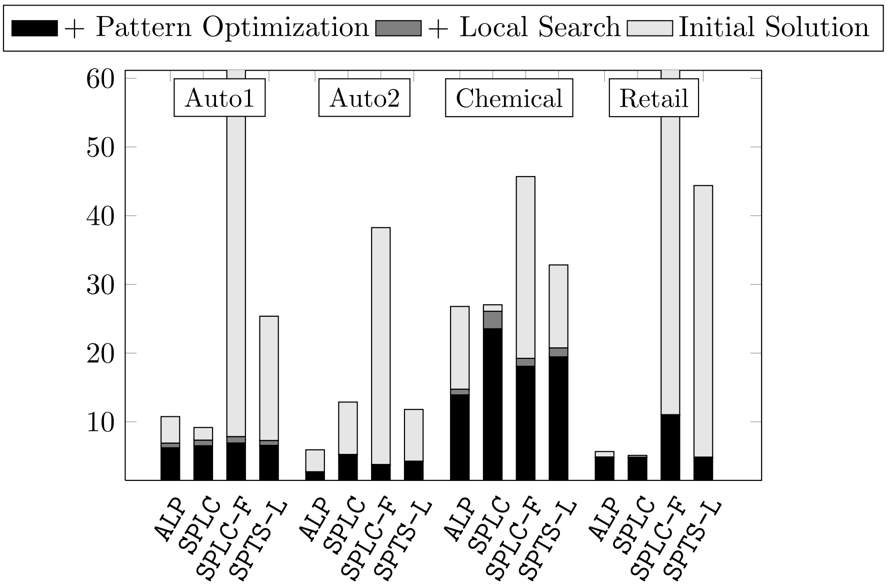
Table 4 shows the gaps of the computed solutions to the lower bound computed by AMIP-B. Throughout the automotive and retail instance sets, the solution quality is within single-digit average gaps to the lower bounds. The local search with LP starting solution and the AMIP-H framework provide the best solution quality, while the performace 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 the multi-commodity 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.

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, on 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 best results, on par with the AMIP-H framework (which takes considerably more computational effort).

Figure 2: Gaps Achieved with Postprocessing in %



Note. The percentaged gap to best known lower bound of fast solvers (time limit 1800 s) for the initial solution, after local search and after pattern optimization are shown—with initial solutions by `SPLC-F` achieving 113% in average on Auto1 and 253% on Retail

Impact of Pattern Optimization

Figure 2 reveals that the effect of pattern optimization is rather weak on the sets Auto2 and Retail, while 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, while the improvement of solution quality due to 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.

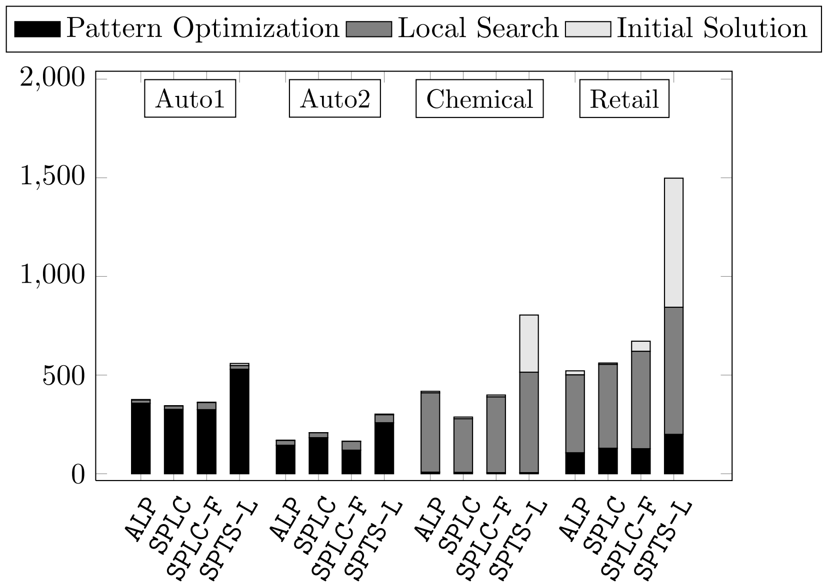
Purely Combinatorial Heuristics

In order 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 heuristic and omitting pattern optimization, the approaches still produce good solutions with a mild loss of at most 3% points of average solution quality.

Comparison with Solutions from Practice

Due to confidentiality reasons 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 customer project has been conducted in cooperation with 4flow AG. The base network consists of 228 consumers, 545 suppliers, 5 hubs, 5857 edges, resulting in a tariff expanded network with 209304 edges in the tariff expanded network. It is fully connected in contrast to the layered structure observed so far. On this instance, the `AMIP-H` framework obtained a solution with 1.2% gap to

Figure 3: Running times of Postprocessing



Note. The table shows shifted geometric means of running times as shares of 2000 s for the local search combined with different algorithms to compute initial solutions and with pattern optimization.

optimality. This constitutes a 14% improvement on the solution obtained with current modelling and algorithmic approaches, which if applied on an annual basis results in savings of up to 1.6 million Euro.

7 Summary & 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 realworld 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 multi-commodity 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 both in connection with standard MIP solvers for optimal solution quality, or as purely combinatorial algorithm, yielding competitive solutions without usage of third-party software. Hence, the broad spectrum of our algorithms offers a flexible trade-off between 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 et al. (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.

Acknowledgements: We 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.

References

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

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

Anily, S., A. Federgruen. 1990. One warehouse multiple retailer systems with vehicle routing costs. *Management Science* **36**(1) 92–114.

Blumenfeld, D.E., L.D. Burns, C.F. Daganzo, M.C. Frick, R.W. Hall. 1987. Reducing logistics costs at general motors. *Interfaces* **17**(1) 26–47.

Bookbinder, J.H., K.E. Reece. 1988. Vehicle routing considerations in distribution system design. *European Journal of Operational Research* **37**(2) 204–213.

Burns, L.D., R.W. Hall, D.E. Blumenfeld, C.F. Daganzo. 1985. Distribution strategies that minimize transportation and inventory costs. *Operations Research* **33**(3) 469–490.

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

Çetinkaya, S. 2005. Coordination of inventory and shipment consolidation decisions: A review of premises, models, and justification. P. M. Pardalos, D. Hearn, J. Geunes, E. Akçali, H. E. Romeijn, Z.-J. M. Shen, eds., *Applications of Supply Chain Management and E-Commerce Research*, *Applied Optimization*, vol. 92. Springer, 3–51.

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

Chan, L.M.A., D. Simchi-Levi. 1998. Probabilistic analyses and algorithms for three-level distribution systems. *Management Science* **44**(11) 1562–1576.

Chopra, S., P. Meindl. 2007. *Supply chain management: strategy, planning, and operations*. Pearson Prentice-Hall.

Chouman, M., T.G. Crainic, B. Gendron. 2009. A cutting-plane algorithm for multicommodity capacitated fixed-charge network design. Tech. rep., Université de Montréal. CRT-2003-16, Centre de recherche sur les transports.

Chuzhoy, J., A. Gupta, J.S. Naor, A. Sinha. 2008. On the approximability of some network design problems. *ACM Transactions on Algorithms (TALG)* **4**(2) 23.

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

Costa, A.M. 2005. A survey on benders decomposition applied to fixed-charge network design problems. *Comput. Oper. Res.* **32** 1429–1450.

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

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

Crainic, T.G., M. Gendreau. 2002. Cooperative parallel tabu search for capacitated network design. *Journal of Heuristics* **8**(6) 601–627.

Crainic, T.G., M. Gendreau, J.M. Farvolden. 2000. A simplex-based tabu search method for capacitated network design. *INFORMS Journal on Computing* **12**(3) 223.

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

Dobson, G. 1982. Worst-case analysis of greedy heuristics for integer programming with nonnegative data. *Mathematics of Operations Research* **7**(4) pp. 515–531.

Fischetti, M., D. Salvagnin, A. Zanette. 2008. Minimal Infeasible Subsystems and Benders Cuts. Tech. rep., University of Padova.

Fischetti, M., D. Salvagnin, A. Zanette. 2010. A note on the selection of benders' cuts. *Math. Program.* **124** 175–182.

Frangioni, A., B. Gendron. 2009. 01 reformulations of the multicommodity capacitated network design problem. *Discrete Applied Mathematics* **157**(6) 1229 – 1241.

Frigge, M., D.C. Hoaglin, B. Iglewicz. 1989. Some implementations of the boxplot. *The American Statistician* **43**(1) pp. 50–54.

Garey, M.R., D.S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York.

Geoffrion, A.M., G.W. Graves. 1974. Multicommodity distribution system design by benders decomposition. *Management Science* **20**(5) 822–844.

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

Ghamlouche, I., T.G. Crainic, M. Gendreau. 2003. Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design. *Operations Research* 655–667.

Ghamlouche, I., T.G. Crainic, M. Gendreau. 2004. Path relinking, cycle-based neighbourhoods and capacitated multicommodity network design. *Annals of Operations research* **131**(1) 109–133.

Jans, R., Z. Degraeve. 2007. Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches. *European Journal of Operational Research* **177**(3) 1855–1875.

Jaruphongsa, W., S. Çetinkaya, C.-Y. Lee. 2005. A dynamic lot-sizing model with multi-mode replenishments: polynomial algorithms for special cases with dual and multiple modes. *IIE Transactions* **37**(5) 453–467(15).

Jaruphongsa, W., S. Çetinkaya, C.-Y. Lee. 2007. Outbound shipment mode considerations for integrated inventory and delivery lot-sizing decisions. *Operations Research Letters* **35**(6) 813–822.

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

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

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

Kliewer, G., L. Timajev. 2005. Relax-and-cut for capacitated network design. *Proceedings of Algorithms-ESA, 2005: LNCS*, vol. 3369. 47–58.

König, F.G., J. Matuschke, A. Richter. 2012. Multi-Dimensional Commodity Covering for Tariff Selection in Transportation. D. Delling, L. Liberti, 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, G.S. 1975. Two NP-complete problems in nonnegative integer programming. Tech. rep., Princeton University. Report No. 178, Computer Science Laboratory.

Magnanti, T.L., R.T. Wong. 1984. Network design and transportation planning: Models and algorithms. *Transportation Science* **18**(1) 1–55.

Richey, MB, R.G. Parker. 1986. On multiple steiner subgraph problems. *Networks* **16**(4) 423–438.

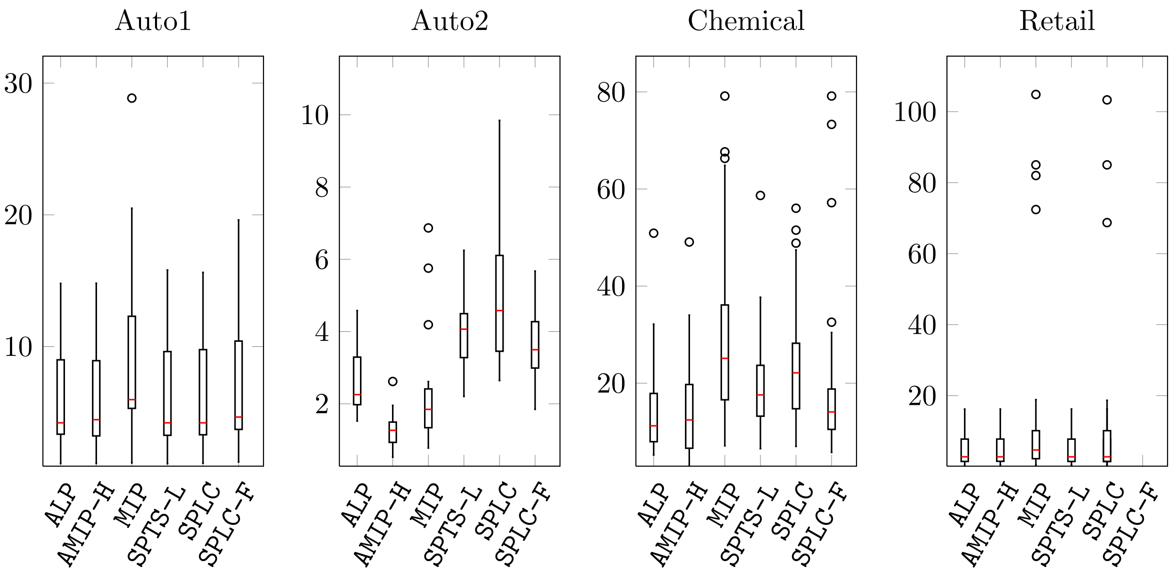
Simchi-Levi, D., P. Kaminsky, E. Simchi-Levi. 2003. *Designing and Managing the Supply Chain: Concepts, Strategies, and Case Studies*. McGraw Hill.

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

Wagner, H.M., T.M. Whitin. 1958. Dynamic version of the economic lot size model. *Management Science* **5**(1) 89–96.

A Tables and Diagrams

Figure 4: Distribution of Solution Quality on all sets

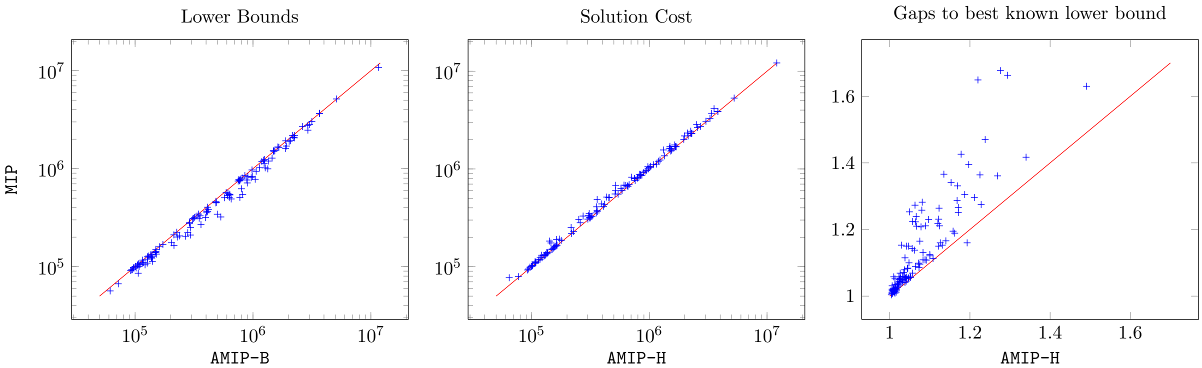


Note. The box and whisker diagrams show the distribution of percentaged gaps to the best known lower bound of the respective method within each test set. The center of the box represents the median of the gaps, the box boundaries represent lower and upper quartiles and the whisker ends show the minimal respectively maximal obtained gap apart from possible outliers that are marked with a cycle (if any). For detailed definitions of quartiles and outliers we followed the suggestions in Frigge et al. (1989).

Overview of Solution Cost and Lower Bounds for exact approaches

The diagrams in Fig. 5 compare the results obtained by branch and bound approaches. We observe that the lower bound obtained by `AMIP-B` are always higher than those of `MIP`. We can also observe that the solution cost obtained by `AMIP-H` is almost always lower than that of `MIP`.

Figure 5: Overview Solution Cost and Lower Bounds for exact approaches



Note. Comparison of `MIP` (ordinate) and `AMIP-B` (abscissa) respectively `MIP` and `AMIP-H` (abscissa) concerning best solution cost and lower bound on each instance.

Detailed Instance Sizes

Tables 5, 6, 7 and 8 show the detailed instance sizes in our test sets.

instance	#nodes in base network			#comm.	#edges		
	#source	#sinks	#facilities		base	pattern exp.	tariff exp.
auto1_1	108	10	127	917	1109	7416	253614
auto1_1QA	60	8	77	349	701	4668	160290
auto1_1QB	13	6	28	153	206	1404	47136
auto1_1QC	47	7	63	196	559	3732	127830
auto1_1QD	22	9	40	218	267	1842	61116
auto1_1QE	8	7	24	214	168	1152	38448
auto1_1QF	18	8	35	136	224	1554	51282
auto1_1SA	30	3	37	282	146	1098	33510
auto1_1SB	25	2	31	71	126	942	28914
auto1_1SC	59	4	67	402	294	2166	67434
auto1_1SD	39	3	51	244	417	2808	95382
auto1_1SE	34	2	40	331	171	1266	39228
auto1_3	92	10	111	348	970	6486	221826
auto1_3QA	47	8	64	129	571	3810	130572
auto1_3QB	12	6	27	50	197	1344	45078
auto1_3QC	35	7	51	79	438	2934	100170
auto1_3QD	20	9	38	93	262	1800	59964
auto1_3QE	8	7	24	73	168	1152	38448
auto1_3QF	17	8	34	53	220	1524	50364
auto1_3SA	29	3	36	107	143	1074	32820
auto1_3SB	18	2	24	32	95	714	21804
auto1_3SC	48	4	56	148	242	1788	55512
auto1_3SD	35	3	47	102	378	2550	86466
auto1_3SE	30	2	36	116	151	1122	34644
auto1_4	92	10	111	257	970	6486	221826
auto1_4QA	47	8	64	105	571	3810	130572
auto1_4QB	12	6	27	35	197	1344	45078
auto1_4QC	35	7	51	70	438	2934	100170
auto1_4QD	20	9	38	63	262	1800	59964
auto1_4QE	8	7	24	47	168	1152	38448
auto1_4QF	17	8	34	42	220	1524	50364
auto1_4SA	29	3	36	71	143	1074	32820
auto1_4SB	18	2	24	27	95	714	21804
auto1_4SC	48	4	56	113	242	1788	55512
auto1_4SD	35	3	47	79	378	2550	86466
auto1_4SE	30	2	36	86	151	1122	34644

Table 5: Auto1 Instance Sizes

instance	#nodes in base network			#comm.	#edges		
	#source	#sinks	#facilities		base	pattern exp.	tariff exp.
auto2_1	64	4	72	383	355	2562	13872
auto2_1A	17	4	24	64	88	672	27054
auto2_1B	29	4	37	188	172	1254	14346
auto2_1C	18	4	25	131	91	696	55812
auto2_1S0511	36	2	42	131	190	1392	29892
auto2_1S0710	42	2	48	256	220	1608	34608
auto2_2	64	4	72	161	355	2562	13872
auto2_2A	17	4	24	27	88	672	27054
auto2_2B	29	4	37	88	172	1254	14346
auto2_2C	18	4	25	46	91	696	55812
auto2_2S0511	36	2	42	64	190	1392	29892
auto2_2S0710	42	2	48	100	220	1608	34608
auto2_3	64	4	72	153	355	2562	13872
auto2_3A	17	4	24	26	88	672	27054
auto2_3B	29	4	37	83	172	1254	14346
auto2_3C	18	4	25	44	91	696	55812
auto2_3S0511	36	2	42	61	190	1392	29892
auto2_3S0710	42	2	48	95	220	1608	34608

Table 6: Auto2 Instance Sizes

instance	#nodes in base network			#comm.	#edges		
	#source	#sinks	#facilities		base	pattern exp.	tariff exp.
chemieTN1	2	844	890	90	38140	234180	1378380
chemieTN2	2	34	48	25	469	3102	17172
chemieTN3	2	59	70	41	613	4098	22488
chemieTN4	2	205	229	64	4776	30030	173310
chemieTN5	2	62	71	36	516	3522	19002
chemieTN6	2	99	109	47	914	6138	33558
chemieTN7	2	122	136	42	1616	10512	58992
chemieTN8	2	58	67	26	483	3300	17790
chemieTN9	2	195	204	56	1596	10800	58680
chemieTN10	6	108	128	88	1741	11214	63444
chemieTN11	6	152	173	120	2573	16476	93666
chemieTN12	6	648	673	243	13318	83946	483486
chemieTN13	6	160	178	79	2204	14292	80412
chemieTN14	6	201	220	105	2961	19086	107916
chemieTN15	6	326	353	107	7400	46518	268518
chemieTN16	6	211	229	84	2860	18534	104334
chemieTN17	6	454	471	154	5694	36990	207810
chemieTN18	1	407	452	29	18359	112866	663636
chemieTN19	3	1247	1294	51	56463	346542	2040432
chemieTN20	3	70	78	23	445	3138	16488
chemieTN21	3	75	92	33	1184	7656	43176
chemieTN22	3	407	434	45	10315	64494	373944
chemieTN23	3	72	84	24	757	5046	27756
chemieTN24	3	124	133	37	908	6246	33486
chemieTN25	3	185	197	34	1903	12600	69690
chemieTN26	3	95	107	31	996	6618	36498
chemieTN27	3	208	216	41	1301	9102	48132
chemieTN28	12	408	453	248	14526	89874	525654
chemieTN29	12	174	209	145	4553	28572	165162
chemieTN30	12	234	274	213	7279	45318	263688
chemieTN31	11	90	124	78	2456	15480	89160
chemieTN32	12	84	115	108	1966	12486	71466
chemieTN33	12	113	147	161	2949	18576	107046
chemieTN34	12	121	151	122	2586	16422	94002
chemieTN35	12	238	272	151	5897	37014	213924
chemieTN36	11	63	96	71	1732	10968	62928
chemieTN37	12	117	149	115	2787	17616	101226
chemieTN38	12	58	90	59	1486	9456	54036
chemieTN39	11	33	64	37	933	5982	33972
chemieTN40	12	84	114	104	1882	11976	68436
chemieTN41	12	997	1041	373	33993	210204	1229994
chemieTN42	12	672	705	269	15551	97536	564066
chemieTN43	12	39	75	36	1290	8190	46890
chemieTN44	12	338	369	205	7209	45468	261738
chemieTN45	12	294	328	201	7224	45312	262032
chemieTN46	11	44	71	45	947	6108	34518
chemieTN47	12	514	548	200	12388	77616	449256
chemieTN48	12	322	364	156	10489	65118	379788
chemieTN49	12	292	330	151	8337	52002	302112
chemieTN50	11	30	66	27	1061	6762	38592

Table 7: Chemical Instance Sizes

instance	#nodes in base network			#comm.	#edges		
	#source	#sinks	#facilities		base	pattern exp.	tariff exp.
handel1	7	702	748	1468	27008	166536	2435208
handel2	7	702	748	316	27008	166536	2435208
handel3	7	702	748	100	27008	166536	2435208
handelTN1	7	60	106	257	2691	16782	242826
handelTN2	7	62	108	143	2485	15558	224298
handelTN3	4	74	117	104	2718	17010	245322
handelTN4	7	75	121	223	3293	20484	297096
handelTN5	7	72	118	182	3144	19572	283668
handelTN6	6	76	121	177	3227	20088	291156
handelTN7	7	65	111	175	2815	17556	254016
handelTN8	7	78	124	234	3162	19716	285324
handelTN9	4	72	115	161	2658	16638	239910
handelTN10	7	68	114	266	2951	18390	266274
handelTN11	2	121	135	142	1605	10440	145260
handelTN12	2	186	198	227	2082	13680	188568
handelTN13	3	269	289	494	5032	31926	454614
handelTN14	2	182	196	210	2401	15582	217266
handelTN15	3	174	193	372	3034	19362	274218
handelTN16	2	212	222	219	1943	12990	176202
handelTN17	2	57	67	78	530	3582	48102
handelTN18	2	62	72	77	581	3918	52722
handelTN19	2	74	87	90	912	5994	82602
handelTN20	2	62	74	118	708	4692	64164
handelTN21	2	72	84	108	820	5424	74304
handelTN22	2	52	64	84	594	3948	53844
handelTN23	3	65	81	121	956	6222	86526
handelTN24	3	78	94	187	1221	7890	110454
handelTN25	3	72	92	159	1381	8838	124842
handelTN26	3	54	70	164	803	5238	72690
handelTN27	3	60	79	176	1085	6984	98124
handelTN28	3	46	65	100	833	5388	75360
handelTN29	3	68	87	180	1212	7794	109602
handelTN30	3	57	81	115	1324	8430	119646
handelTN31	3	72	96	88	1652	10488	149256
handelTN32	3	65	89	116	1496	9510	135174
handelTN33	7	359	405	669	13952	86142	1258110
handelTN34	7	343	389	911	13323	82272	1201404
handelTN35	7	344	390	900	13948	86028	1257660
handelTN36	7	494	540	1214	19720	121560	1778040
handelTN37	4	413	456	724	15966	98532	1439676
handelTN38	3	340	382	744	12971	80118	1169682

Table 8: Retail Instance Sizes