Branch-and-Price-and-Cut for a Service Network Design and Hub Location Problem

Ann-Kathrin Rothenbächer*,[a], Michael Drexl[a,b], Stefan Irnich[a]

[a]*Chair of Logistics Management, Johannes Gutenberg University Mainz, Jakob-Welder-Weg 9, D-55128 Mainz, Germany.*
[b]*Fraunhofer Centre for Applied Research on Supply Chain Services SCS, Nordostpark 93, D-90411 Nuremberg, Germany.*

Abstract

In the context of combined road-rail freight transport, we study the integrated tactical planning of hub locations and the design of a frequency service network. We consider a number of real-world constraints such as multiple transshipments of requests at hubs, transport time limits for requests, request splitting, and outsourcing possibilities. To our knowledge, the combination of problem features we deal with has not been described before. We present a path-based model and solve it with a branch-and-price-and-cut algorithm. Computational experiments show that large realistic instances from a major German rail freight company can be solved close to optimality within one hour on a standard desktop computer, allowing our algorithm to be used for practical planning purposes.

Key words: Service Network Design, Hub Location, Intermodal Transport, Branch-and-Price-and-Cut

1. Introduction

In this paper, we consider a real-world integrated tactical service network design and hub location problem (SNDHLP) for combined transport by road and rail. Combined transport is a special kind of intermodal transport. The latter is defined by the Economic Commission for Europe (2001) as the movement of freight in one and the same loading unit, such as a container or swap body, which uses successively two or more modes of transport without handling the freight itself in changing modes. Combined road-rail transport is characterized by the additional condition that the main part of the transport is by rail, and only the initial and the final leg are performed by road, see Figure 1. Intermodal and combined transport are economically attractive because of the consolidation effects they offer and the resulting potential for cost savings and efficiency gains. Combined transport traffic has more than doubled in the European Union since 1990. In 2011, about 6.4 million containers were transported with a freight performance of 44.7 billion tonne-kilometres. Combined transport has thus become a well-established sector of freight transport, with a market encompassing dozens of companies generating a joint turnover of several billion Euros per year (European Commission, 2014). A further aspect of combined transport is its presumed environment-friendliness. Although this view is not undisputed (Dekker *et al.*, 2012), combined transport is commonly considered more ecologically beneficial than pure road transport (Craig *et al.*, 2013), and the European Union and national governments have set up programmes for fostering combined transport (Council of the European Communities, 1992). In short, planning problems in combined transport are of considerable significance for a large number of enterprises as well as the economy and society as a whole.

The SNDHLP we consider is as follows. We are given a set of transport requests. Each request consists of one or more containers that must be transported from an origin location to a destination location. Each location may be the origin and/or the destination of more than one request, but all containers with the

*Corresponding author.
Email address: `anrothen@uni-mainz.de` (Ann-Kathrin Rothenbächer)

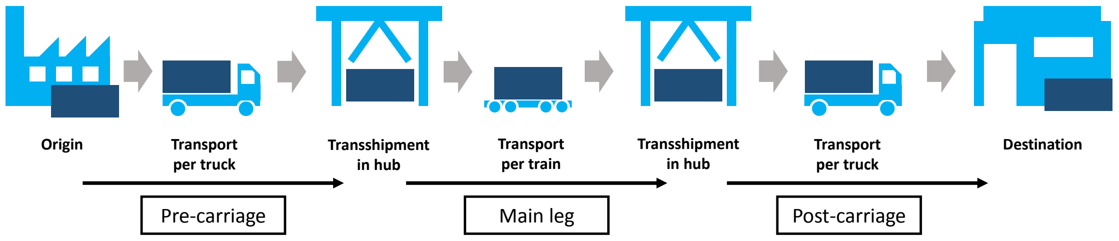

Figure 1: Schematic structure of combined transport

same origin and the same destination constitute one request. There are two possibilities for transporting a container: combined road-rail transport and direct transport from origin to destination by road. In the former case, the number of *hops*, i.e., the number of times a container may be transshipped at hubs, is limited to at most four. This includes the transshipments from road to rail at the start hub and from rail to road at the end hub. Combined transport is relevant only for requests beyond a certain origin-destination distance, so that, for any request, the start and the end hub used are always different, and the number of hops is always at least two. There is a maximal time en route, including travelling and transshipment, for each request. This time is assumed non-restrictive in the case of road transport. The decision whether to transport a request by combined transport or entirely by road is part of the problem.

There are two types of hubs, *fixed* and *potential* ones. Fixed hubs are open and available. In addition, a limited number of the potential hubs may be opened and used. Because of the considered tactical planning horizon, no fixed costs are incurred for using or not using a fixed hub or for opening and using a potential hub. The decision which hubs to use is part of the problem. Each request can enter the rail network, i.e., be transshipped from road vehicle to rail wagon, at a given request-specific set of start hubs. Similarly, a request can leave the rail network, i.e., be transshipped from rail wagon to road vehicle, at a given request-specific set of end hubs. For any request, these two sets are disjoint, and they may each contain one element only. The selection of the start and the end hub for each request that uses combined transport is part of the problem.

We consider two problem variants, with *unsplittable* and *splittable* requests. 'Unsplittable' means that all containers belonging to the same request must take the same itinerary. 'Splittable' means that the itineraries of two containers from one request may be different (including different start and end hubs), and that some containers from one request may be transported by road and some by combined transport.

Owing to the tactical nature of the problem, the capacities of the rail links are assumed nonrestrictive. This means that at any point in time, an unlimited number of trains may use a rail link without affecting the travel time of a train. The number of trains to operate between an ordered pair of hubs is part of the problem. Similarly, the transshipment capacities at the hubs are assumed nonrestrictive, too. This means that an unlimited number of requests can be transshipped at a hub without any requests having to wait for transshipment. Moreover, we assume homogeneous equipment, i.e., there is only one type of container, road vehicle, and rail wagon. In particular, we assume that each available road vehicle and rail wagon can carry only one container at a time. The number of wagons, i.e., the capacity of a train, is limited and identical for all rail connections. The cost of a train trip between two hubs is independent of the actual capacity utilization of the train, i.e., of the number of rail wagons actually pulled. Costs for direct road transport are known for each request. Hence, from the point of view of a combined transport operator, direct road transport can be seen as outsourcing. The cost and time for transshipping one container are hub-specific, but at each hub, they are the same for all requests.

Overall, the problem is to decide on the itineraries of the requests, the usage of hubs, and the number of trains operating between hubs, with the objective of minimizing the overall costs for fulfilling all requests, either by combined transport or by pure road transport. These overall costs are comprised of the costs for operating trains between hubs and of the road transport costs. In the well-known classification of network design problems by Crainic (2000), the SNDHLP we consider is a *frequency service network design* problem, because it focuses on determining which traffic itineraries to operate and how often over the planning horizon to offer a train service. Scheduling aspects referring to when services depart, which are the subject of *dynamic service network design*, are not considered. Additionally, our problem possesses a hub location component, because a restricted number of hubs to use must be selected from a larger set.

The contributions of the present paper consist in the development of (i) a path-based mixed-integer programming model for the SNDHLP, and (ii) a branch-and-price-and-cut algorithm capable of solving instances with up to 100 requests and 10 potential hubs to optimality. For real-world instances with up to 5300 requests and 43 hubs, feasible solutions with a gap of less than 2 % are obtained within 60 minutes. This allows to use our implementation for practical planning purposes. Moreover, our solution approach consistently outperforms a state-of-the-art commercial solver applied out-of-the-box to an arc flow model.

The remainder of this paper is organized as follows. The next section provides a review of literature on hub location and service network design problems related to the SNDHLP. Section 3 presents our path-based formulation of the SNDHLP. In Section 4, the branch-and-price-and-cut algorithm is explained. Computational results are presented in Section 5, and we give a conclusion in Section 6.

2. Literature review

Assad (1980) was the first to present a survey of planning problems in rail transport. Further reviews were given by Crainic and Laporte (1997), Crainic (2002), and, especially for intermodal transport, by Bontekoning *et al.* (2004), Crainic *et al.* (2006), Caris *et al.* (2013), and SteadieSeifi *et al.* (2014). The problem we consider addresses both service network design and hub location aspects. Often, these two topics are treated independently. Wieberneit (2008) and Yaghini and Akhavan (2012) gave reviews of service network design problems occurring in freight transport. A good overview of hub location problems can be found in Alumur and Kara (2008). The simultaneous consideration of hub location and request routing decisions, as in our SNDHLP, increases the complexity of the models, but makes much more efficient solutions possible. There are several problem variants studied in the literature. In the following, we briefly describe important variants and then refer chronologically to papers dealing with problems that share properties with the one we study. We did not find a paper dealing with the same combination of problem features as our SNDHLP. A summarizing comparison of papers with respect to the considered problem aspects and the applied solution procedures is provided in Table 1.

As mentioned, requests can either be enforced to be transported as a whole (unsplittable requests) or the loading units of a request can use different routes (splittable requests). If several requests starting or ending at the same location have to be assigned to the same hub, this is called *single allocation*, whereas the opposite is called *multiple allocation*. Most papers model economies of scale caused by consolidation with the help of a discount factor smaller than 1, which is multiplied with the transport costs incurred for the use of links between two hubs. In this case, there is no reason for transshipments when a fully connected network is considered. A more accurate treatment determines the number of required means of transport to provide the needed capacity for each link individually. This causes discretely increasing costs and thereby a significantly higher complexity (Crainic, 2000). With this discrete capacity determination of the links, transshipping at hubs between means or modes of transport can improve the utilization and should be taken into account, though this also increases the difficulty. Furthermore, as in our problem, some papers view the option of a direct transport from origin to destination at potentially higher costs, and there are also papers considering bounds for the transport times.

The first to consider hub location and routing of goods simultaneously was O'Kelly (1986). The positions of the hubs were not chosen out of a discrete set but located freely in the plane. Only the cases of one and two hubs were considered. Campbell (1994) gave several formulations for different variants of a hub location problem, regarding explicitly the decision of the usage of links. Brockmüller *et al.* (1996) studied a network design problem arising in the telecommunication sector, addressing the decision over integral numbers of capacities to be installed on the arcs. The problem was solved heuristically with the help of valid inequalities. Barahona (1996) considered a similar problem with the same application regarding both splittable and unsplittable flows. Yoon and Current (2008) were the first to study an integrated hub location

Table 1: Characteristics of SNDHLP and of similar models from the literature

	Our paper	Brockmüller *et al.* (1996)	Barahona (1996)	Yoon and Current (2008)	Campbell (2009)	Bauer *et al.* (2010)	Ishfaq and Sox (2011)	Üster and Agrahari (2011)	Alumur *et al.* (2012)	Homfeld (2012)	Contreras and Fernández (2012)	Zhang *et al.* (2013)
Network design	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
Hub location	p-median			Fixed opening costs	p-median	p-median	p-median, fixed costs	Fixed opening costs	p-median, fixed costs		p-median	
Application	Intermodal transport	Telecommunication network	Telecommunication network	Not specified	Time definite motor carriers	Intermodal transport	Intermodal transport road-rail	Freight transport	Multimodal transport	Rail freight transport	Not specified	Intermodal transport road-rail-inland waterway
Direct transport possible	✓	✓		✓		✓	✓					✓
Splittable requests	✓		✓	✓	✓	✓		✓				
Unsplittable requests	✓	✓	✓			✓	✓		✓	Extended, see text	✓	✓
Multiple allocation	✓	✓	✓	✓	✓	✓	✓	✓				✓
Discretely increasing link costs	✓	✓	✓					✓		✓		
Maximal number of hops	4						2	2		9		
Transport time limit	✓				✓		✓		✓	✓		
Solution approach	Branch-and-price-and-cut	Mat-heuristic	Heuristic	Dual ascent heuristic	MIP solver	MIP solver with static cuts	Tabu search	Benders decomposition	Mat-heuristic	Mat-heuristic	MIP solver	Heuristic bilevel programming

and network design problem similar to our setting. However, they decided only on the usage of arcs and not on arc capacities. The first author looking at transport time limits in the context of hub location and network design was Campbell (2009). Ishfaq and Sox (2011) also addressed time limits. They used the context of combined road-rail transport and supported the possibility to choose between both modes of transport on each arc. Bauer *et al.* (2010), in a follow-up to earlier work by Andersen *et al.* (2009), studied a problem similar to our SNDHLP. They considered two different objective functions: (i) the minimzation of greenhouse gas emissions, disregarding any costs, and (ii) the minimization of the total costs of providing the services plus the sum of the time-en-route of all requests weighted by corresponding values of time. They developed an arc flow model based on a time-space network. Using a commercial solver and two types of valid inequalities, they were able to solve small real-world instances with up to 15 requests and 5 hubs to optimality. Bauer *et al.* (2010) compared the results obtained with the two objective functions and concluded that the cost minimization version leads to more services being offered, so that the capacity utilization of the services is higher when emissions are minimized. Another problem closely related to our setting was regarded by Üster and Agrahari (2011). While looking at the same decisions, they restricted themselves to the case of splittable requests and assumed that only one long-haul link is used by each request. They solved the problem with Benders decomposition. In his doctoral thesis, Homfeld (2012) addressed rail freight services with explicit decisions on the number of trains running on each link between hubs. He looked at a special hierarchical order of consolidation centres and considers the requirement that two requests with the same destination have to use the same remaining route, once they meet at a hub. By their own account, the most general form of a hub location and network design problem addressed in the literature was regarded by Alumur *et al.* (2012). They allowed multiple modes of transport on each link. Contreras and Fernández (2012) studied the simultaneous optimization of hub locations and network design under the additional requirement that the chosen arcs result in a tree, connecting all hubs with a unique path between them. Zhang *et al.* (2013) studied the optimization of intermodal networks considering road, rail, and inland waterway, taking into account costs for CO_2 emissions. As in our problem, they considered multiple commodities and request-specific sets of start and end hubs. However, they did not consider splittable requests, time-en-route limits, or transport economies of scale through consolidation. They solved their problem with heuristic bilevel programming. The upper level, which is solved by a genetic algorithm, searches for hubs to be opened and adequate CO_2 emission prices. The lower level, which is solved by a shortest-path algorithm for each request, computes minimal transport and CO_2 costs given a certain network structure and CO_2 prices.

3. Mathematical model

Let $G = (V, A)$ be a digraph with node set V and arc set A. The node set V contains one origin and one destination node for each request, and one node for each hub. The set of all hub nodes is denoted by H, and the set of fixed hubs by H_f, with H_f a subset of H. The number of hubs that have to be used is given by n_H. The set of arcs contains the rail arcs between any two hubs, A_t, with costs per kilometre c_t and the road arcs, A_s, with costs per kilometre c_s. The road arcs are composed of the direct road arcs from the origins to the destinations of the requests and the arcs connecting the origins and destinations to the hub network. Each request $r \in R$ comprises m^r containers. The length of an arc a is given by l_a. The train capacity is k_t.

We tackle the problem with a path-based model. A feasible combined transport path for a request is an elementary path from the origin of the request to the destination, passing through an allowed start and an allowed end hub, performing at most the allowed number of hops, and maintaining the travel time constraint. A road transport path for a request is simply the single-arc path from the origin of the request to the destination and is feasible by definition. The set of feasible paths for a request r is denoted by P^r. P_i and P_a indicate the sets of paths containing hub i and arc a respectively, with $P_i^r = P_i \cap P^r$ and $P_a^r = P_a \cap P^r$.

The model contains three types of decision variables: Binary variables h_i indicate whether hub $i \in H$ is used. General integer variables t_a count the number of trains that use arc $a \in A_t$. Continuous variables $y_p^r \in [0, 1]$ indicate the fraction of the containers of request $r \in R$ which are transported via path $p \in P^r$.

Given these definitions, the SNDHLP can be formulated as follows:

$$\begin{aligned}
\min \quad & \sum_{a\in A_t} c_t l_a t_a + \sum_{r\in R}\sum_{p\in P^r}\sum_{a\in p\cap A_s} c_s l_a m^r y_p^r && && \text{(1a)}\\
\text{s.t.} \quad & h_i = 1 \qquad \forall\, i \in H_f && && \text{(1b)}\\
& \sum_{i\in H} h_i = n_H && && \text{(1c)}\\
& \sum_{p\in P^r} y_p^r = 1 \qquad \forall\, r \in R && (\gamma_r) && \text{(1d)}\\
& \sum_{r\in R}\sum_{p\in P_a^r} m^r y_p^r \le k_t t_a \qquad \forall\, a \in A_t && (\delta_a) && \text{(1e)}\\
& \sum_{p\in P_i^r} y_p^r \le h_i \qquad \forall\, i \in H, r \in R && (\epsilon_i^r) && \text{(1f)}\\
& h_i \in \{0,1\} \qquad \forall\, i \in H && && \text{(1g)}\\
& t_a \in \mathbb{N}_0 \qquad \forall\, a \in A_t && && \text{(1h)}\\
& y_p^r \in [0,1] \qquad \forall\, r \in R, p \in P^r && && \text{(1i)}
\end{aligned}$$

The objective function, (1a), seeks to minimize the total costs incurred by the covered rail and road kilometres. Conditions (1b) and (1c) ensure that all fixed hubs are opened and that the required number of hubs is used. Constraint (1d) makes sure that every request is transported. Constraint (1e) calculates the used capacity and determines the number of needed trains per arc. By constraint (1f), a hub is forced to be opened if a path passes through it. The last three constraints determine the domains of the variables.

The above model targets the case of splittable requests, but the following simple domain modification ensures that all containers of a request are transported over the same path:

$$y_p^r \in \{0,1\} \qquad \forall\, r \in R, p \in P^r$$

The problem can also be formulated with arc variables. Then, the feasibility of the routes concerning transport times and limits for the number of hops has to be expressed explicitly in the model. For the unsplittable case, this is straightforward, whereas for the splittable case, additional time and hop counter variables are needed. Linear formulations for these models can be found in the Appendix.

4. Branch-and-Price-and-Cut

The number of feasible paths can be huge, so an enumeration of all possible routes is not practicable. Instead, we use delayed column generation to dynamically generate promising paths. To achieve integrality, this is embedded in branch-and-bound. Besides, we use valid inequalities to strengthen the linear relaxation.

To solve our path-based SNDHLP model (1), we developed a branch-and-price-and-cut algorithm, see Barnhart *et al.* (1996), Desaulniers *et al.* (2005), and Lübbecke and Desrosiers (2005). In the following subsection, we illustrate our pricing algorithm to find new paths with negative reduced costs. Afterwards, we present our branching scheme and some valid inequalities, for which we describe their influence on the pricing problem. At the end of this section, we mention some techniques that enhance the performance of the branch-and-price-and-cut algorithm.

4.1. Generation of new columns

For the column generation, we consider the linear relaxation of the master problem. To have a feasible problem initially, we start with a small subset of paths, containing, for each request, one path that represents the complete transport by road. Recall that the pricing problem searches for variables with negative reduced costs to add them to the current set of columns of the master problem. Let γ_r be the dual variable associated to constraint (1d) for request $r \in R$, δ_a the dual variable corresponding to the capacity restriction (1e) of arc $a \in A$, and ϵ_i^r the dual belonging to constraint (1f) for hub $i \in H$ and request $r \in R$. Then, the reduced costs for a path p for request r are

$$\tilde{c}_p = \sum_{a \in p \cap A_s} c_s m^r l_a - \gamma_r - \sum_{a \in p \cap A_t} m^r \delta_a - \sum_{i \in p \cap H} \epsilon_i^r. \tag{2}$$

The subproblem for a certain request $r \in R$ is a shortest path problem with resource constraints (SPPRC). SPPRCs are well studied, because they appear in many applications, see Irnich and Desaulniers (2005). They are usually solved by dynamic-programming based labelling algorithms. One of the first algorithms in this category is the one proposed by Aneja *et al.* (1983). More recent papers compare mono- and bi-directional extensions (Righini and Salani, 2006) and deal with special properties of such algorithms in the context of branch-and-price-and-cut algorithms (Zhu and Wilhelm, 2013).

Let $r \in R$ be a request for which we solve the pricing problem. We define a label $l = (v, \bar{l}, z)$ that marks a path from the origin to a node $v \in V$ with $\bar{l}$ as the previous label on the path and $z \in \mathbb{R}_+^m$ as the vector of resource consumptions of the path. For our purposes, the resource vector $z = (z_1, z_2, z_3) = (\tilde{c}, d, h)$ contains the sum of the modified arc costs $\tilde{c}$, the transport time d and the number of hops h of the partial path up to the associated node. As most dual variables are zero, particularly at the beginning of the column generation process, and as shorter paths tend to be cheaper and are more likely to be part of an optimal solution, it seems reasonable to search for paths with the most negative reduced costs and, among these, the one with the shortest transport time and the lowest number of hops. Therefore, we consider a lexicographic order of objectives with bounds for the downstream targets. All labels belonging to the same node can be compared according to the order of the objectives $(\tilde{c} - d - h)$:

Definition 4.1 (Order of labels)**.** A label $l = (v, \bar{l}, z)$ is better than a label $l' = (v, \bar{l}', z')$, written $l < l'$, if

$$\exists\ i \in \{1, 2, 3\} : (z_j = z'_j\ \forall\ j < i) \wedge (z_i < z'_i).$$

Besides, we say a label dominates an other label belonging to the same node, if the former is at least as good as the other label in all resource categories and strictly better in at least one resource consumption:

Definition 4.2 (Dominance between labels)**.** A label $l = (v, \bar{l}, z)$ dominates a label $l' = (v, \bar{l}', z')$, written $l \prec l'$, if

$$z \leq z' \wedge \exists\ i \in \{1, 2, 3\} : z_i < z'_i.$$

In general, an SPPRC can be solved with a monodirectional or a bidirectional algorithm. Because of the special structure respecting the limited hop-number, we implemented both a monodirectional forward and a bidirectional pricing routine in order to compare them. In the former case, we extend partial paths only from the origin, always continuing with the best unprocessed label as in the Dijkstra algorithm. In the latter case, we start from both the origin and the destination, make a predefined number of hops from both sides, determine all non-dominated paths, and choose the best alternative from a concluding merge step.

In the following, we describe the forward extension of the paths. The dual prices corresponding to the hub usage, the transshipment times on the hubs, and the increase of the hop counter are assigned to the ingoing arcs. Starting from a label l, we look at all outgoing arcs to other hubs, add the weight and resource consumptions for this path extension, and create a new label. Let the hub indicator ζ_i be 1 if $i \in H$ and 0 otherwise, and let the arc indicator ξ_a be 1 if $a \in A_t$ and 0 if $a \in A_s$. For the sake of readability, we write ξ_{ij} for $\xi_{(i,j)}$ (similar for other variables). With d_a as the transport time over arc a and d_i as the transshipment time on hub i, the new label is determined as follows:

$$l_{\text{new}} = (j,\ l, \tilde{c}(l_{\text{new}}),\ d(l_{\text{new}}),\ h(l_{\text{new}})) \tag{3}$$

with

$$\tilde{c}(l_{\text{new}}) = \tilde{c}(l) + (1 - \xi_{ij}) c_s m^r l_{ij} - \xi_{ij} m^r \delta_{ij} - \zeta_j \epsilon_j^r \tag{4a}$$

$$d(l_{\text{new}}) = d(l) + d_{ij} + \zeta_j d_j \tag{4b}$$

$$h(l_{\text{new}}) = h(l) + \zeta_j. \tag{4c}$$

Then we test whether this is feasible with regard to the resource limits. In addition, we check whether a feasible extension of the new label to the destination exists. Afterwards, we check whether the new label is dominated by one of the labels already assigned to the node. If yes, it is discarded, otherwise it is added to the label list of the node. Finally, all other labels are tested for dominance by the new one and are deleted in this case. We store the label list of a node as an ordered list, so that insertion and dominance are performed simultaneously and take time linear in the number of labels at the node, see Algorithm 1. The backward extension of paths works analogically.

Algorithm 1: Potential insertion of a new label

```
Input: An ordered label list $list$ belonging to a node, a new label $l_{\text{new}}$
$l = list$.first();
while $l < l_{new}$ do
    if $l \prec l_{new}$ then
        Delete $l_{\text{new}}$;
        return;
    $l = list$.next();
Insert $l_{\text{new}}$ before $l$;
while $l \neq list.end()$ do
    if $l_{new} \prec l$ then
        Delete $l$;
    $l = list$.next();
```

The monodirectional pricer is applied on the network defined in Section 3. For the bidirectional pricer, we use a layered network constructed as follows: If $u_{\max}$ is the maximal allowed number of hops, we copy the hubs $u_{\max} + 1$ times to create $u_{\max} + 1$ hub node layers, numbered from 1 to $u_{\max} + 1$. The first (last) layer contains only copies of hubs in the request-specific set of start (end) hubs. The bidirectional pricer propagates labels forward from the origin vertex only up to layer $\lfloor u_{\max}/2 \rfloor + 1$, and backward from the destination vertex only up to layer $\lfloor u_{\max}/2 \rfloor + 2$. Hence, forward (backward) road arcs are inserted from the origin (destination) vertex to all copies in the first (last) layer, and forward (backward) rail arcs are inserted between any two copies of different hubs in neighbouring layers from layer 1 up to layer $\lfloor u_{\max}/2 \rfloor + 1$ (from layer $u_{\max}$ back to layer $\lfloor u_{\max}/2 \rfloor + 2$). This yields a network as depicted in Figure 2, on which the shortest path search is done in both directions.

In this approach, we store label lists for each hub on each layer, so that the dominance criterion only needs to consider the modified arc costs and the transport time of the labels, but not the hop counter resource. To take into account all possible numbers of hops $(2, \dots, u_{\max})$, we merge the label lists of all hubs at the last backward layer ($\lfloor u_{\max}/2 \rfloor + 2$) with all forward layers except the first (1) and the label lists of the first forward layer with all backward layers except the first ($u_{\max} + 1$). The exceptions are motivated by the requirement that start and end hub of a request path are always different, as mentioned in the Introduction.

4.2. Branching

For a feasible solution, the hub variables (h) must be binary, the number of trains running between any two hubs (t) must be integer, and, in the case of unsplittable requests, also the path variable values (y) must be binary. To achieve this, we use the following branching scheme.

4.2.1. Branching on hub variables

Using or not using a hub has the biggest influence on the solution, so these decisions are taken first. To identify the most important hub, we calculate the number of containers transshipped at each hub in the

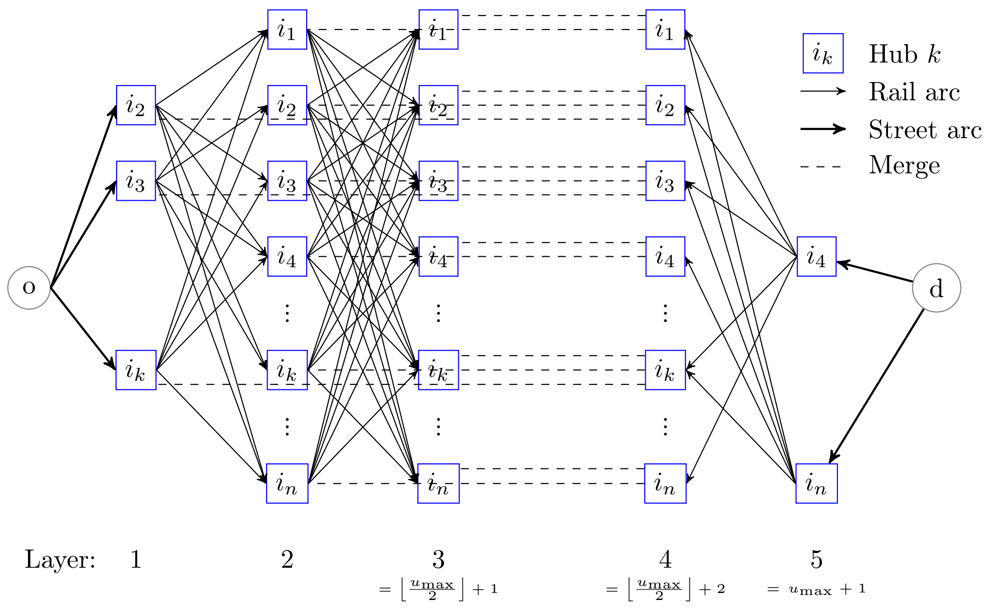

Figure 2: Layered network for the bidirectional pricing algorithm, $u_{\max} = 4$

current LP solution. For the hub i with the largest such number and fractional h_i, we create two branches, one fixing the value of the corresponding variable h_i to 0 and the other one to 1 by changing the upper and lower bounds, respectively. Whereas the 1-branch has no impact on the pricing algorithm, the 0-branch requires arcs ending at the corresponding hub to be excluded from the network.

4.2.2. Branching on train variables

For the number of trains, we first use a global approach, regarding the total number of trains running in the whole network, n_t. Whenever n_t is fractional, we create two branches and add a constraint to the master problem in both branches, setting the upper bound on the number of trains to $\lfloor n_t \rfloor$ in one branch, and setting the lower bound to $\lceil n_t \rceil$ in the other. This requires no change in the pricing problems. To guarantee integrality of the train variables on each arc, we have to branch further by setting similar bounds on individual train variables. If the upper bound of a train variable is set to zero, the corresponding arc has to be removed from the network of the pricing problem. In all other cases, the branching decision does not affect the pricing problem. For splittable requests, the described branching rules suffice to obtain a feasible solution.

4.2.3. Branching on path variables

In the case of unsplittable requests, variables generated in the pricing routine cannot simply be forbidden by setting their upper bound to zero, because the excluded path will be regenerated by the pricing problem solver. We devised two approaches to treat integrality requirements on path variables, both concerning single arcs of a path.

The first possibility directly regards single arcs. To forbid an arc, all paths containing this arc are excluded from the network, and the pricing problem ignores this arc. The branch that enforces the arc a for request r, however, needs an additional constraint in the master problem:

$$\sum_{p \in P_a^r} y_p^r \geq 1.$$

The dual variable corresponding to this constraint takes nonnegative values, which can cause negative arc weights in the subproblem. To avoid cycles in the resulting paths, the algorithm searching for the best new variable entering the master problem would have to be adapted.

We therefore chose a second possibility, one that does not change the nature of the pricing problem. If u indicates the number of hubs in a path, this path will contain $u+1$ *stages*. The first stage consists of the arc from the origin to the first hub, the $(u+1)$th stage is the arc from the uth hub to the destination. The second possibility to branch on path variables now uses the idea to forbid and enforce arcs for use on certain stages of a path for a request. In the branch in which an arc is forbidden, the upper bound of all paths using this arc on the chosen stage is set to zero, and the arc is excluded from the pricing algorithm at this stage. In the other branch, in which the arc is enforced to be used by this request on the requested stage, all path variables not fulfilling this requirement are set to zero, and the pricing problem excludes all other arcs on this stage.

4.3. Cuts

To strengthen the linear relaxations of the master problem, we use several types of valid inequalities that are based on the main source of fractionality, the capacity restriction (1e). With $b_r = \frac{m^r}{k_t}$, this constraint for an arc a can be rewritten as

$$\sum_{r\in R} b_r \sum_{p\in P_a^r} y_p^r \le t_a.$$

Of course, because of the positive costs for the trains, this inequality will always be fulfilled with equality in the LP relaxation, resulting in mostly fractional train variable values. Moreover, we see that not individual paths, but the sum of all path variables, i.e., the total flow, for the same request containing an arc has to be considered. In the following, we present three cuts with different application areas.

4.3.1. Simple cuts

The simplest valid inequality, adapted from the *strong cuts* described in (Chouman *et al.*, 2003), is only useful for arcs with light traffic, i.e., for those arcs with train variable values of less than one. In those cases, we can add the following cut:

$$t_a \ge \sum_{p\in P_a^r} y_p^r \quad \forall\, r \in R_a. \tag{5}$$

The separation of these cuts is done by inspection: We look at all train variables with values in $(0,1)$, compare them with the flows of the requests over this arc, and add all violated ones. In the pricing routine, these cuts need to be incorporated through additional arc costs. Of course, the dual price has to be added only for the request and the arc that are addressed by the simple cut.

4.3.2. Residual capacity cuts

For splittable requests, we use the so called *residual capacity cuts*, developed by Magnanti *et al.* (1993). For a subset $S \subseteq R$, define the sum $B_S := \sum_{r\in S} b_r$. Then, the cuts can be written as

$$(B_S - \lfloor B_S\rfloor)t_a + B_S - (B_S - \lfloor B_S\rfloor)\lceil B_S\rceil \ge \sum_{r\in S} b_r \sum_{p\in P_a^r} y_p^r. \tag{6}$$

To get the most violated inequality for an arc a, Atamtürk and Rajan (2002) showed that one has to choose the set $S = \left\{ r\in R : \sum_{p\in P_a^r} y_p^r > t_a - \lfloor t_a\rfloor \right\}$. Then, it suffices to check whether the current solution satisfies

$$\lfloor t_a\rfloor < B_S < \lceil t_a\rceil \quad \text{and}$$

$$\sum_{r\in R} b_r(1 - \sum_{p\in P_a^r} y_p^r - \lceil t_a\rceil + t_a) < (\lceil t_a\rceil - t_a)\lfloor t_a\rfloor.$$

If so, the corresponding residual capacity cut can be added to the master problem. Otherwise, no inequality of this type is violated by the current LP solution for the considered arc. Let μ_{Sa} be the dual price for a residual capacity cut represented by (S,a). Then, in the pricing algorithm for a request r, the weight $b_r\mu_{Sa}$ has to be added to the arc costs of an arc a, if $r\in S$.

4.3.3. c-Strong cuts

In the case of unsplittable requests, a stronger valid inequality can be used. Brockmüller *et al.* (1996) presented the so called *c-strong-inequality* in the application area of telecommunication networks and Homfeld (2012) applied them to a rail freight transport problem.

For a $c \in \mathbb{N}_0$, a set $S \subset R$ is called c-strong if $c = \sum_{r \in R} \lceil b_r \rceil - \lceil \sum_{r \in R} b_r \rceil$. For such a c-strong set S, a valid inequality for our problem for unsplittable requests can be written as

$$t_a + c \geq \sum_{r \in S} \lceil b_r \rceil \sum_{p \in P_a^r} y_p^r + \sum_{r \in R_a \setminus S} (\lceil b_r \rceil - 1) \sum_{p \in P_a^r} y_p^r. \tag{7}$$

To separate these inequalities, the following knapsack problem has to be solved:

$$\begin{aligned} \max \quad & \sum_{r \in R_a} \pi_r \sum_{p \in P_a^r} y_p^r \\ \text{s.t.} \quad & \sum_{r \in R_a} (\lceil b_r \rceil - b_r)\pi_r < c + 1 \\ & \pi_r \in \{0, 1\} \qquad \forall\, r \in R_a \end{aligned}$$

Then, the set $S = \{r \in R_a \mid \pi_r = 1\}$ yields the most violated c-strong cut for the arc a. The consideration of these cuts in the subproblem solver is similar to above. Let λ_{Sa} be the dual variable belonging to a c-strong cut with request-subset S and arc a. Then all requests in S must add $\lceil b_r \rceil \lambda_{Sa}$ to the modified costs of arc a, whereas all other requests have to add the rounded-down variant $\lfloor b_r \rfloor \lambda_{Sa}$.

4.4. Algorithmic improvements

In this section, we mention some techniques we use to improve our branch-and-price-and-cut algorithm. These are partial pricing, dynamic constraint generation, early branching, and usage of a MIP solver to obtain upper bounds.

4.4.1. Partial pricing

To accelerate the branch-and-price-and-cut algorithm, we perform partial pricing. Instead of searching for a negative reduced cost path for all requests in each iteration, we sometimes restrict ourselves to the most promising ones: Requests for which an entering path variable was found in the last pricing iteration. The reason is that these requests are more likely to yield another negative reduced cost path in the following pricing step. Nevertheless, single path changes can cause completely different dual variables and, hence, there may also be other requests with improving columns in later iterations. Thus, with probability

$$1 - \frac{\text{Number of requests with new path in last iteration}}{\text{Total number of requests}},$$

we search for new columns only for the most promising requests. Naturally, when the partial pricing does not generate new columns, a full search has to be done to guarantee optimality.

4.4.2. Dynamic constraint generation

The number of constraints ensuring that hubs on used paths are opened can be very high (number of hubs · number of requests) (1f). Because this can make the model unnecessarily large, we generate them on demand. When no more improving paths are found, we enumerate all hub-request pairs and add the corresponding constraint (1f) if it is violated. Since one requests suffices to enforce a hub to be opened, we limit the number of such constraints added in one iteration.

4.4.3. Early branching

Although in general the cuts significantly increase the lower bound, they enlarge the model and herewith increase the solution time for the LP relaxation. Sometimes, also the branching decision has a considerable impact on the lower bound. Therefore, we restrict the separation of cuts in two ways. Firstly, we set a fixed bound on the minimal violation required for a cut to be added to the model. Secondly, we limit the number of cut-generation iterations per branch-and-bound node.

4.4.4. Usage of a MIP solver

Our branch-and-bound algorithm works with a best-first branching strategy, so that always the node with the smallest lower bound is explored next. Thus, there is no need for a good upper bound to reject nodes early, as those nodes with lower bounds above the optimal solution are simply fathomed as soon as the latter one is found. Nevertheless, the branch-and-price-and-cut algorithm may stop because of a timeout without having found an optimal or even a feasible solution. To improve the chances of obtaining a good feasible solution in such a case, we apply a MIP solver on the restricted problem (1) with the current set of columns at different stages of the algorithm: After solving the root node, we apply the MIP solver with a running time limited to 60 seconds. In the case of a timeout, we start the MIP solver again. Because the model is much larger at this point (compared to the root node) and the best found solution directly decreases the optimality gap, we allow a running time of five minutes.

5. Computational study

In this section, we present computational experiments to analyze the effectiveness of the proposed branch-and-price-and-cut procedure. The algorithm was implemented in C++, compiled with Microsoft Visual Studio 2010, and used the CPLEX 12.6 library for solving linear programs and as a MIP solver as described in the previous section. The experiments were performed on a standard PC with an Intel Core i7-4790 CPU at 3.60 GHz, 8 GB of RAM, running Windows 7 Enterprise.

5.1. Instances

Europe's largest railway company, DB Mobility Logistics AG, kindly provided us with two sets of requests based on real-life data with different container-per-request ratios. The sets respectively contain 3,300 and 5,300 shipments inside Germany, suitable for combined transport, and 43 hub locations, 18 of which are considered fixed. By randomly choosing subsets of requests and selecting the hubs with the largest capacity, we generated smaller instances. The hubs selected in this way were still evenly spread throughout Germany. Furthermore, we varied the maximal allowed number of hubs to be used. In total, we generated 144 smaller instances, ranging from 100 to 500 requests and from 10 to 21 hubs. The request-specific sets of allowed start and end hubs were constructed by considering all hubs that were within 1/6 of the Euclidean origin-destination distance of a request. This resulted in an average of 2.5 potential start and end hubs per request for the largest hub set. For 21 and 10 hubs, the averages were 1.9 and 1.1, respectively. The maximal number of hops was set to $u_{\max} = 4$. The time-en-route limits were calculated by multiplying the road transport time with a factor of between 1.5 and 2.5, increasing with growing origin-destination distances. The ratio of the costs per kilometer on street to the costs per kilometer on rail was assumed to be 3/7. All test instances were tackled by our algorithm regarding the requests as splittable as well as unsplittable.

5.2. Algorithmic settings

We tried several algorithmic variants to solve the problem. Table 2 shows a comparison of different settings: (i) use of bidirectional or monodirectional pricer, (ii) branching on the sum of all trains, (iii) using the cuts from Section 4.3, and (iv) dynamically adding constraints (1f) to the model. The results are based on six instances with 100 requests and 21 potential hubs from which 10 have to be chosen, and on six instances with 500 requests and 21 potential hubs from which 15 have to be chosen, with a runtime of 15 minutes. The deviations listed in the table demonstrate clearly that the cutting planes presented in Section 4.3 have a significant impact on the lower bound. The benefits of the dynamically generated constraints are smaller, but appear in all instances. Furthermore, the results indicate that branching on the sum of all used trains does not improve the lower bound. The choice of the best pricing algorithm depends on the instance; the bidirectional pricer produces slightly better lower bounds on average. Therefore, we decided to choose as 'best setting' the one with the bidirectional pricing algorithm, without branching on the sum of all trains, with the use of all described cuts, and with dynamic generation of constraints (1f).

Table 2: Comparison of lower bounds for different solving strategies

		Use bidirectional pricer					
		true	true	false	false		
Branch on sum of trains	true	1.57 (1.71)	49.34 (49.71)	1.51 (1.72)	49.54 (49.66)	true	Generate dynamic constraints
	true	1.69 (1.91)	50.22 (51.13)	1.75 (2.10)	50.51 (50.77)	false	
	false	0.13 (0.11)	42.20 (42.70)	0.22 (0.10)	42.33 (42.48)	true	
	false	0.15 (0.38)	42.94 (43.29)	0.38 (0.29)	43.22 (43.95)	false	
		true	false	true	false		
		Use cuts					

The values show the sum (over the twelve test instances used) of the relative deviations of the lower bound of this setting from the best lower bound found among all settings for splittable requests (and, in parentheses, for unsplittable requests).

5.3. Results

To assess the capabilities of our branch-and-price-and-cut algorithm, we compared its performance, using the best setting mentioned above, to that of the standard solver Cplex, applied out-of-the-box to arc-based formulations for the SNDHLP (specified in the Appendix). The findings were unequivocal: For all test instances, our algorithm yielded much better lower bounds, up to 86 % above those obtained by Cplex, and 23.5 % higher on average. Both our algorithm and Cplex had no problems in finding feasible solutions, but the upper bounds from Cplex were, on average, 27 % worse for splittable and 14.2 % worse for unsplittable requests. This justifies the modelling and implementation efforts spent on the path-based formulation and the branch-and-price-and-cut algorithm.

We then applied our algorithm (with the best setting) to all instances described in Section 5.1. Tables 3 and 4 present the results. The time limit for the large instances was set to one hour, whereas for the smaller instances, 30 minutes were allowed. Table 3 indicates, for each large instance, the solution time (where 'T.L.' indicates that the time limit was reached), the relative gap between the final lower bound and the best found feasible solution, the percentage of the overall computation time needed for cut separation, pricing, and reoptimization, the number of added cuts, the number of pricing runs, the number of generated paths, the number of solved branch-and-bound nodes, and the number of requests that were transported completely by road in the best integer solution. In Table 4, the same information is displayed for each group of smaller instances. In addition, the number of instances with a gap of less than 3 % is indicated.

The most important insights are the following:

- Our algorithm was able to solve to optimality instances with up to 100 requests and 21 hubs. For almost all instances, an optimality gap in the single-digit area was reached within one hour.
- Of the 36 instances with 100 requests, 18 split (19 unsplit) could be solved to optimality. For 500 requests, only 6 (10) of the 36 instances finished with a gap greater than 3 %. Of the 12 large instances, 2 (4) could be solved with a gap of less than 1 %, and only three had a gap of more than 3 %.
- Naturally, with an increasing number of requests, the problem becomes harder, and fewer instances were solved to optimality. The same happened with growing numbers of hubs, even though there is no decision on hub usage when all potential hubs have to be opened to achieve the requested number of hubs. This is because more hubs allow disproportionately more connections and thus a higher number of train variables.
- A small number of potential hubs permitted the solution of much more branch-and-bound nodes, as both the pricing algorithm and the reoptimization consumed less time. On the other hand, the number of paths found was smaller, and much more requests were transported entirely by road in solutions to such instances.
- The difference between settings with splittable and unsplittable requests was insignificant. Both the

Table 3: Results for large instances

Instance	$\|H\|$	n_H	Time in s	%Gap	%Separation time	%Pricing time	%Reopt. time	#Cuts	#Pricings	#Paths	#Nodes	#Only Road
5392-Split	43	18	T.L.	0.28	0.6	72.7	2.4	2219	426	95427	93	1705
3351-Split	43	18	T.L.	0.26	1.6	65.8	4.2	3577	757	34746	369	1552
5392-Split	43	20	T.L.	2.58	0.7	71.4	1.6	2708	379	80812	101	689
3351-Split	43	20	T.L.	5.26	1.3	68.2	4.6	4501	671	35020	315	1130
5392-Split	43	25	T.L.	6.31	0.4	78.0	3.2	4710	395	81214	63	279
3351-Split	43	25	T.L.	12.82	0.7	66.7	17.0	7131	591	41494	163	692
5392-Unsplit	43	18	T.L.	0.33	0.8	63.7	2.9	1696	664	99787	113	1699
3351-Unsplit	43	18	T.L.	0.58	1.9	56.8	5.2	2846	973	34129	447	1549
5392-Unsplit	43	20	T.L.	0.85	0.8	70.7	2.1	2149	780	89925	112	677
3351-Unsplit	43	20	T.L.	0.92	1.6	63.8	4.6	3806	1068	37656	395	1137
5392-Unsplit	43	25	T.L.	1.80	0.4	80.8	2.9	3894	901	79626	69	272
3351-Unsplit	43	25	T.L.	1.92	0.9	63.9	15.6	5572	1106	42158	219	616

Table 4: Average results for small instances, 6 instances per group

Instance group	$\|H\|$	n_H	Time in s	%Gap	#Gap $< 3\%$	%Separation time	%Pricing time	%Reopt. time	#Cuts	#Pricings	#Paths	#Nodes	#Only Road
100-Split	10	5	373	0.00	6/6	0.1	44.5	42.0	933	2405	2856	1754	44
100-Split	10	10	851	0.86	6/6	0.1	33.9	49.4	2357	2602	2788	1961	2
100-Split	21	5	636	0.14	6/6	0.1	28.2	65.7	1658	2033	12484	288	57
100-Split	21	10	1263	1.34	5/6	0.2	21.2	71.5	3575	2507	12022	458	27
100-Split	21	15	1289	3.31	3/6	0.2	16.6	75.8	4920	2135	8761	570	12
100-Split	21	21	T.L.	6.00	2/6	0.2	13.4	77.3	5946	2671	8525	922	7
500-Split	10	5	T.L.	0.31	6/6	0.2	41.3	38.0	1770	4945	11632	3900	223
500-Split	10	10	T.L.	1.56	6/6	0.2	65.5	16.1	3964	3723	5953	3387	5
500-Split	21	5	T.L.	0.45	6/6	0.1	8.8	86.6	1835	1392	40195	157	293
500-Split	21	10	T.L.	2.12	4/6	0.1	15.7	77.5	2714	1444	28879	281	128
500-Split	21	15	T.L.	2.38	6/6	0.1	23.2	68.2	4635	1166	18903	304	43
500-Split	21	21	T.L.	14.47	2/6	0.1	28.9	63.5	7956	773	11700	218	7
100-Unsplit	10	5	665	0.00	6/6	0.1	31.1	40.5	769	4142	2893	3459	44
100-Unsplit	10	10	1041	1.12	5/6	0.1	26.4	42.4	1925	4184	2809	3504	2
100-Unsplit	21	5	634	0.20	6/6	0.1	26.0	68.0	1416	2058	12616	295	57
100-Unsplit	21	10	1256	2.51	3/6	0.1	17.8	73.4	2983	2681	12504	497	27
100-Unsplit	21	15	1270	4.29	2/6	0.1	13.8	80.2	4025	2300	8998	645	13
100-Unsplit	21	21	T.L.	10.14	2/6	0.2	10.5	81.9	4734	3038	8892	1146	5
500-Unsplit	10	5	T.L.	0.29	6/6	0.3	34.1	40.3	1494	5323	11668	4265	226
500-Unsplit	10	10	T.L.	1.84	6/6	0.3	59.4	19.5	3178	4913	6347	4542	4
500-Unsplit	21	5	T.L.	0.54	6/6	0.1	6.9	88.2	1504	1469	41146	165	292
500-Unsplit	21	10	T.L.	1.67	6/6	0.1	12.5	80.5	2241	1607	30054	325	127
500-Unsplit	21	15	T.L.	4.31	2/6	0.1	18.6	72.4	3742	1384	19730	367	44
500-Unsplit	21	21	T.L.	10.80	0/6	0.1	26.4	65.8	6617	994	12368	321	6

solution times and the solution values were very similar, and a lot of instances that could be solved to optimality even had the same solution. This is probably caused by the fact that there are only few requests with a large number of containers relative to the train capacity (there are only 38 requests overall with a number of containers exceeding 25 % of the train capacity).

6. Conclusions

We have studied a specific variant of an integrated tactical service network design and hub location problem for combined road-rail transport, considering several practically relevant constraints. We have developed a branch-and-price-and-cut algorithm using different pricing algorithms, tailored branching strategies, and problem-specific cutting planes. Computational experiments with instances using real-world data show that our path-based algorithm clearly and consistently outperforms a commercial solver using an arc-based model. Our algorithm is capable of finding close-to-optimal solutions in less than one hour of computation time, even for instances of realistic size, and can thus be applied for practical planning.

Further research could be done with respect to model extensions and with respect to algorithmic enhancements. As for model extensions, balancing constraints for locomotives and wagons could be added, as well as capacities at hubs and fixed costs for hub usage and train services. Focusing on a model and solution approach for solving a concrete real-world problem, we have abstained from doing so. However, in different application contexts, these aspects may be relevant, and they can rather easily be integrated into our model.

With regard to algorithmics, our procedure could be improved by developing a primal heuristic that runs faster than the current approach of solving an extended set-covering problem with existing columns using a MIP solver. Furthermore, a method for which successful applications to network design and hub location problems are reported in the literature is Benders decomposition. Therefore, applying this technique to the problem described in this paper constitutes an interesting research avenue.

Acknowledgments

We thank Hanno Schülldorf from Deutsche Bahn Mobility Logistics AG for providing us the realistic network and request data.

During earlier work on the problem described in this paper, M. Drexl was partially funded by the German Federal Ministry of Education and Research (BMBF) within the project ‘E-Motion’ under grant no. 05M13ANA. He thanks the BMBF for the opportunity to participate in the project E-Motion.

Appendix: Arc-based models

Here, we describe the arc-based models for the unsplit and the split request variants of our SNDHLP. We need the following additional notation: Let d^r_{max} be the maximal allowed transport time, and let A^r_{start} and A^r_{end} be the possible street arcs connecting the origin and destination of request r to the hub network. Then, the problem proposed in this paper can be formulated with an arc-based approach as follows:

Arc-based model with unsplittable requests

The following variables are used:

- $h_i \in \{0,1\}$ indicates whether or not hub $i \in H$ is used
- $t_a \in \mathbb{N}_0$ counts the number of trains running on arc $a \in A_t$
- $x^r_a \in \{0,1\}$ decides whether or not request $r \in R$ uses arc $a \in A$

The model is then:

$$\min \quad \sum_{a\in A_t} c_t l_a t_a + \sum_{r\in R}\sum_{a\in A_s} c_s m^r l_a x^r_a \tag{8a}$$

$$\text{s.t.} \quad h_i = 1 \qquad \forall\, i \in H_f \tag{8b}$$

$$\sum_{i\in H} h_i = n_H \tag{8c}$$

$$\sum_{a\in A^r_{\text{start}}} x^r_a = 1 \qquad \forall\, r \in R \tag{8d}$$

$$\sum_{a\in A^r_{\text{end}}} x^r_a = 1 \qquad \forall\, r \in R \tag{8e}$$

$$\sum_{a\in A^r\cap\delta^-(i)} x^r_a - \sum_{a\in A^r\cap\delta^+(i)} x^r_a = 0 \qquad \forall\, r \in R, i \in H \tag{8f}$$

$$\sum_{r\in R} m^r x^r_a \le k_t t_a \qquad \forall\, a \in A_t \tag{8g}$$

$$\sum_{a=(i,j)\in A^r} (d_a + d_j) x^r_a \le d^r_{\text{max}} \qquad \forall\, r \in R \tag{8h}$$

$$\sum_{a\in A_t} x^r_a \le u_{\text{max}} - 1 \qquad \forall\, r \in R \tag{8i}$$

$$x^r_a \le h_i \qquad \forall\, r \in R, a = (i,j) \in A_t \tag{8j}$$

$$x^r_a \le h_j \qquad \forall\, r \in R, a = (i,j) \in A_t \tag{8k}$$

$$h_i \in \{0,1\} \qquad \forall\, i \in H \tag{8l}$$

$$t_a \in \mathbb{N}_0 \qquad \forall\, a \in A_t \tag{8m}$$

$$x^r_a \in \{0,1\} \qquad \forall\, r \in R, a \in A^r \tag{8n}$$

The minimization objective (8a) sums the costs for the covered kilometres. Constraints (8b) and (8c) respectively ensure that the fixed hubs are opened and that the correct number of hubs is available. The next three constraints ensure flow conservation. The train capacities are maintained by constraints (8g). Constraints (8h) and (8i) limit the total transport time and the number of transshipments for each request. Constraints (8j) and (8k) ensure that a hub is open if a request traverses it, and the last three constraints describe the domains of the variables.

Arc-based model with splittable requests

The following variables are used:

- $h_i \in \{0,1\}$ indicates whether or not hub $i \in H$ is used
- $t_a \in \mathbb{N}_0$ counts the number of trains running on arc $a \in A_t$
- $x_a^r \in [0,1]$ describes the percentage of request $r \in R$ transported over arc $a \in A$
- $u_a^r \in \{0,1\}$ indicates whether or not a part of request $r \in R$ uses arc $a \in A$
- $m_i^r \in \mathbb{N}_0$ measures the time en route of a part of request $r \in R$ upon reaching hub $i \in H$
- $n_i^r \in \mathbb{N}_0$ counts the number of transshipments that a part of request $r \in R$ has performed up to hub $i \in H$

The model is then:

$$\min \quad \sum_{a\in A_t} c_t l_a t_a + \sum_{r\in R}\sum_{a\in A_s} c_s m^r l_a x_a^r \tag{9a}$$
$$\text{s.t.} \quad h_i = 1 \qquad \forall\, i \in H_f \tag{9b}$$
$$\sum_{i\in H} h_i = n_H \tag{9c}$$
$$\sum_{a\in A^r_{\text{start}}} x_a^r = 1 \qquad \forall\, r \in R \tag{9d}$$
$$\sum_{a\in A^r_{\text{end}}} x_a^r = 1 \qquad \forall\, r \in R \tag{9e}$$
$$\sum_{a\in A^r\cap\delta^-(i)} x_a^r - \sum_{a\in A^r\cap\delta^+(i)} x_a^r = 0 \qquad \forall\, r \in R, i \in H \tag{9f}$$
$$\sum_{r\in R} m^r x_a^r \le k_t t_a \qquad \forall\, a \in A_t \tag{9g}$$
$$x_a^r \le u_a^r \qquad \forall\, r \in R, a \in A_t \tag{9h}$$
$$(d_a + d_j) u_a^r \le m_j^r \qquad \forall\, r \in R, a = (i,j) \in A^r_{\text{start}} \tag{9i}$$
$$m_i^r + (d_a + d_j) u_a^r \le m_j^r + d^r_{\max}(1 - u_a^r) \qquad \forall\, r \in R, a = (i,j) \in A_t \tag{9j}$$
$$m_i^r + d_a u_a^r \le d^r_{\max} \qquad \forall\, r \in R, a = (i,j) \in A^r_{\text{end}} \tag{9k}$$
$$u_a^r \le n_j^r \qquad \forall\, r \in R, a = (i,j) \in A^r_{\text{start}} \tag{9l}$$
$$n_i^r + u_a^r \le n_j^r + u_{\max}(1 - u_a^r) \qquad \forall\, r \in R, a = (i,j) \in A_t \tag{9m}$$
$$n_i^r \le u_{\max} \qquad \forall\, r \in R, a = (i,j) \in A^r_{\text{end}} \tag{9n}$$
$$u_a^r \le h_i \qquad \forall\, r \in R, a = (i,j) \in A_t \tag{9o}$$
$$u_a^r \le h_j \qquad \forall\, r \in R, a = (i,j) \in A_t \tag{9p}$$
$$h_i \in \{0,1\} \qquad \forall\, i \in H \tag{9q}$$
$$t_a \in \mathbb{N}_0 \qquad \forall\, a \in A_t \tag{9r}$$
$$x_a^r \in [0,1] \qquad \forall\, r \in R, a \in A^r \tag{9s}$$
$$u_a^r \in \{0,1\} \qquad \forall\, r \in R, a \in A^r \tag{9t}$$
$$m_i^r \in \mathbb{N}_0 \qquad \forall\, r \in R, i \in H \tag{9u}$$
$$n_i^r \in \mathbb{N}_0 \qquad \forall\, r \in R, i \in H \tag{9v}$$

Again, the objective is to minimize the total costs for the covered kilometres. The next constraints, (9b)–(9g), are also the same as above. Constraint (9h) determines the correct values for the binary flow variables on the basis of the fractional flow variables. The following three constraints ensure that the time en route of a request upon reaching a certain node is set correctly, and that the time at the destination does not exceed the allowed maximal time en route. This requirement is split into the first road leg (9i), the rail legs in between (9j), and the last road leg (9k). The same is done for the number of transshipments per request on each hub in the following three constraints, (9l)–(9n). Then, the relation between the hub and path variables is expressed as in the unsplittable case above by constraints (9o) and (9p). The last six constraints regulate the domains of the variables.

References

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

Alumur, S. A., Kara, B. Y., and Karasan, O. E. (2012). Multimodal hub location and hub network design. *Omega*, **40**(6), 927–939.

Andersen, J., Crainic, T. G., and Christiansen, M. (2009). Service network design with management and coordination of multiple fleets. *European Journal of Operational Research*, **193**(2), 377–389.

Aneja, Y. P., Aggarwal, V., and Nair, K. P. K. (1983). Shortest chain subject to side constraints. *Networks*, **13**(2), 295–302.

Assad, A. A. (1980). Models for rail transportation. *Transportation Research Part A*, **14A**, 205–220.

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

Barahona, F. (1996). Network design using cut inequalities. *SIAM Journal on Optimization*, **6**(3), 823–837.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., and Vance, P. H. (1996). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, **46**, 316–329.

Bauer, J., Bektaş, T., and Crainic, T. G. (2010). Minimizing greenhouse gas emissions in intermodal freight transport: An application to rail service design. *Journal of the Operational Research Society*, **61**(3), 530–542.

Bontekoning, Y., Macharis, C., and Trip, J. (2004). Is a new applied transportation research field emerging?–A review of intermodal rail-truck freight transport literature. *Transportation Research Part A*, **38**, 1–34.

Brockmüller, B., Günlük, O., and Wolsey, L. (1996). Designing private line networks – polyhedral analysis and computation. Technical report, Université Catholique de Louvain, Belgium.

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

Campbell, J. F. (2009). Hub location for time definite transportation. *Computers & Operations Research*, **36**(12), 3107–3116.

Caris, A., Macharis, C., and Janssens, G. K. (2013). Decision support in intermodal transport: A new research agenda. *Computers in Industry*, **64**(2), 105–112.

Chouman, M., Crainic, T. G., and Gendron, B. (2003). A cutting-plane algorithm based on cutset inequalities for multicommodity capacitated fixed charge network design. Technical Report CRT-2003-16, Centre de la recherche sur les transports.

Contreras, I. and Fernández, E. (2012). General network design: A unified view of combined location and network design problems. *European Journal of Operational Research*, **219**(3), 680–697.

Council of the European Communities (1992). *Council Directive 92/106/EEC of 7 December 1992 on the establishment of common rules for certain types of combined transport of goods between Member States.* Available at http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:01992L0106-20130701&from=EN. Accessed 20th May 2015.

Craig, A., Blanco, E., and Sheffi, Y. (2013). Estimating the co_2 intensity of intermodal freight transportation. *Transportation Research Part D*, **22**, 49–53.

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

Crainic, T. G. (2002). *A Survey of Optimization Models for Long-haul Freight Transportation.* Centre de Recherche sur les Transports, Montreal, Canada.

Crainic, T. G. and Laporte, G. (1997). Planning models for freight transportation. *European Journal of Operational Research*, **97**, 409–438.

Crainic, T. G., Kim, K. H., *et al.* (2006). Intermodal transportation. *Transportation*, **14**, 467–537.

Dekker, R., Bloemhof, J., and Mallidis, I. (2012). Operations research for green logistics – an overview of aspects, issues, contributions and challenges. *European Journal of Operational Research*, **219**(3), 671–679.

Desaulniers, G., Desrosiers, J., and Solomon, M., editors (2005). *Column Generation.* Springer, New York.

Economic Commission for Europe (2001). *Terminology on Combined Transport.* Available at http://www.unece.org/fileadmin/DAM/trans/wp24/documents/term.pdf. Accessed 10th April 2015.

European Commission (2014). *EU Transport in Figures – Statistical Pocketbook 2014.* Available at http://ec.europa.eu/transport/facts-fundings/statistics/doc/2014/pocketbook2014.pdf. Accessed 13th June 2015.

Homfeld, H. (2012). *Consolidating Car Routes in Rail Freight Service by Discrete Optimization.* Ph.D. thesis, Technische Universität Darmstadt.

Irnich, S. and Desaulniers, G. (2005). Shortest path problems with resource constraints. In G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors, *Column Generation*, pages 33–65. Springer, New York.

Ishfaq, R. and Sox, C. R. (2011). Hub location-allocation in intermodal logistic networks. *European Journal of Operational Research*, **210**(2), 213–230.

Lübbecke, M. E. and Desrosiers, J. (2005). Selected topics in column generation. *Operations Research*, **53**(6), 1007–1023.

Magnanti, T., Mirchandani, P., and Vachani, R. (1993). The convex hull of two core capacitated network design problems. *Mathematical Programming*, **60**(1-3), 233–250.

O'Kelly, M. E. (1986). The location of interacting hub facilities. *Transportation Science*, **20**, 92–105.

Righini, G. and Salani, M. (2006). Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optimization*, **3**(3), 255–273.

SteadieSeifi, M., Dellaert, N., Nuijten, W., Woensel, T. V., and Raoufi, R. (2014). Multimodal freight transportation planning: A literature review. *European Journal of Operational Research*, **233**(1), 1–15.

Üster, H. and Agrahari, H. (2011). A benders decomposition approach for a distribution network design problem with consolidation and capacity considerations. *Operations Research Letters*, **39**(2), 138–143.

Wieberneit, N. (2008). Service network design for freight transportation: A review. *OR Spectrum*, **30**(1), 77–112.

Yaghini, M. and Akhavan, R. (2012). Multicommodity network design problem in rail freight transportation planning. *Procedia – Social and Behavioral Sciences*, **43**(0), 728–739.

Yoon, M.-G. and Current, J. (2008). The hub location and network design problem with fixed and variable arc costs: Formulation and dual-based solution heuristic. *The Journal of the Operational Research Society*, **59**(1), pp. 80–89.

Zhang, M., Wiegmans, B., and Tavasszy, L. (2013). Optimization of multimodal networks including environmental costs: A model and findings for transport policy. *Computers in Industry*, **64**(2), 136–145.

Zhu, X. and Wilhelm, W. E. (2013). Implementation of a three-stage approach for the dynamic resource-constrained shortest-path sub-problem in branch-and-price. *Computers & Operations Research*, **40**(1), 385–394.