Gutenberg School of Management and Economics

& Research Unit "Interdisciplinary Public Policy"

Discussion Paper Series

Schedule-Based Integrated Inter-City Bus Line Planning for Multiple Timetabled Services via Large Multiple Neighborhood Search

Konrad Steiner

December 20, 2018

Discussion paper number 1902

Johannes Gutenberg University Mainz
Gutenberg School of Management and Economics
Jakob-Welder-Weg 9
55128 Mainz
Germany
https://wiwi.uni-mainz.de/

Contact details

Konrad Steiner
A.T. Kearney GmbH
Dreischeibenhaus 1
40211 Düsseldorf
Germany
Chair of Logistics Management
Gutenberg School of Management and Economics
Johannes Gutenberg University Mainz
Jakob-Welder-Weg 9
55128 Mainz
Germany

ksteiner@uni-mainz.de

Schedule-Based Integrated Inter-City Bus Line Planning for Multiple Timetabled Services via Large Multiple Neighborhood Search

Konrad Steiner[*,a,b]

[a]A.T. Kearney GmbH, Dreischeibenhaus 1, D-40211 Düsseldorf, Germany.
[b]Chair of Logistics Management, Gutenberg School of Management and Economics, Johannes Gutenberg University Mainz, Jakob-Welder-Weg 9, D-55128 Mainz, Germany.

Abstract

This work addresses line planning for inter-city bus networks, which requires a high level of integration with other planning steps. One key reason is given by passengers choosing a specific timetabled service rather than just a line, as is typically the case in urban transportation. Schedule-based modeling approaches are required to incorporate this aspect, i.e., demand is assigned to a specific timetabled service. Furthermore, in liberalized markets, there is usually fierce competition within and across modes. This encourages considering dynamic demand, i.e., not relying on static demand values, but adjusting them based on the trip characteristics.

We provide a schedule-based mixed-integer model formulation allowing a bus operator to optimize multiple timetabled services in a travel corridor with simultaneous decisions on both departure time and which stations to serve. The demand behaves dynamically with respect to departure time, trip duration, trip frequency, and cannibalization. To solve this new problem formulation, we introduce a large multiple neighborhood search (LMNS) as an overall metaheuristic approach, together with multiple variations including matheuristics. Applying the LMNS algorithm, we solve instances based on real-world data from the German market. Computation times are attractive and the high quality of the solutions is confirmed by analyzing examples with known optimal solutions. Moreover, we show that the explicit consideration of the dependencies between the different timetabled services often produces insightful new results that differ from approaches which only focus on a single service.

Key words: integration, schedule-based modeling, inter-city bus transportation, dynamic demand, large multiple neighborhood search LMNS

1. Introduction

The planning problem of designing a public transport system is highly complex and has not yet been solved by a fully integrated approach. Traditionally, the problem has been tackled by a sequential planning process (e.g. Desaulniers and Hickman, 2007; Ibarra-Rojas *et al.*, 2015). In the first step, the physical network is designed based on an expected demand profile. This is followed by the selection of a line plan and frequencies. After that, a timetable is determined, which then

[*]Corresponding author.
Email address: `ksteiner@uni-mainz.de` (Konrad Steiner)

serves as a base for the operational planning steps vehicle scheduling, crew scheduling, and crew rostering.

In line planning for inter-city bus transportation, a high level of integration with other planning steps is required. One key reason is given by passengers choosing a specific timetabled service rather than just a line, as is typically the case in urban transportation. As a consequence, demand modeling is linked with timetabling aspects. The modeling approach of assigning demand to specific timetabled services is referred to as *schedule-based modeling*. On top of this, in liberalized markets, there is usually fierce competition within and across modes. This encourages considering *dynamic demand*, i.e., not relying on static demand values, but adjusting the demand based on the trip characteristics. This approach considers aspects such as sensitivity to travel times and cannibalization explicitly.

While the schedule-based nature of demand has been considered on a predictive level in several studies (e.g., in Cascetta and Coppola, 2016), prescriptive approaches are rare. In (Steiner and Irnich, 2018), the authors present a schedule-based model allowing a bus operator to optimize a single timetabled service in a travel corridor. The model simultaneously decides on departure time and which stations to serve. Dynamic demand is considered in two ways: First, different times of the day show different levels of demand to reflect typical travel patterns. Second, the number of possible passengers depends on the duration of a trip, i.e., if there are more intermediate stations between two cities, the demand for the trip will be lower. In this paper, we present an extension of this model that can select multiple timetabled services simultaneously and considers interdependencies between them.

Cities along a travel corridor can vary significantly in size, and therefore have different service frequency requirements. As a consequence, we do not require every selected timetabled service to stop at the exact same stations. Yet, this creates a need for also considering dynamic demand effects that result from the different structures of the individual timetabled services. When developing demand models in practice, we found that the two most important aspects for a pair of stations s_i and s_j are *trip frequency* and *cannibalization*, hence these are considered in the model we present.

We refer to the total number of timetabled services stopping at both stations s_i and s_j during the planning period in scope (e.g., one day) as *trip frequency* for the stations s_i and s_j. The trip frequency impacts the demand for a specific trip from s_i to s_j in two ways. On the one hand, higher trip frequencies increase overall attractiveness of the operator's offer and thus increase the demand: Customers who prefer to travel with this operator and check the offered trips of this operator first, are more likely to find a suitable service. On the other hand, there is also a negative effect of higher trip frequencies because passengers who travel with this operator anyway and are more flexible with respect to the departure time can now distribute between more services. It is not clear a priori which of these effects dominates the other. In fact, this depends on the specific trip frequencies, stations, level of competition, customer groups, and further application-specific aspects. In any case, we consider it favorable for a decision support model to capture these trip frequency effects on the demand.

If two trips between the stations s_i and s_j are offered by the same operator with departure times close to each other, a *cannibalization* effect can be observed for the demand. Specifically, passengers who would have taken either trip, can clearly only take one trip in case both are offered. Again, it is not trivial to determine when departure times of such trips can be considered "close", nor how big these cannibalization effects will be. Yet, providing a network planning model covering this aspect allows for linking with more sophisticated and potentially non-linear demand models.

Altogether, we present a schedule-based mixed-integer linear model that allows us to determine optimal stations and departure times for multiple timetabled services simultaneously. The model includes a many-to-many demand structure which behaves dynamically with respect to departure time, trip duration, trip frequency, and cannibalization. Note that the methods to generate high-quality demand forecasts are not in scope of this paper, they are discussed briefly from a practical perspective in Section 6 of (Steiner and Irnich, 2018) and more fundamentally in (Ortúzar and Willumsen, 2011).

The problem formulation and the input data are based on an example from a German inter-city bus carrier. Requirements and constraints of actual operations have been considered in defining the modeling scope. However, due to the recent consolidation in the German inter-city bus market (e.g. Fockenbrock and Heide, 2017), the collaboration was brought to an end before the model could be applied in the regular planning process.

While the existing model for single timetabled services from (Steiner and Irnich, 2018) allows for exact solutions in acceptable computation times, we doubt that the same can be achieved in this extended context. This is due to the significant increase in model size caused by additionally considering the dynamic demand effects with respect to trip frequency and cannibalization. Hence, we present approaches based on metaheuristics and matheuristics for this purpose. We introduce a large multiple neighborhood search (LMNS) as an overall metaheuristic approach. This is motivated by the successful application of metaheuristics from the LNS family to similar problems, which we discuss further in Section 2.2. Also, the structure of solutions allows for an intuitive definition of operators adjusting existing timetabled services or stations within services. Further, having already developed an optimization model and solution algorithm for single timetabled services, we analyze whether efficient matheuristics based on this model can be designed. The structure of solutions fits well with the general decomposition approach of matheuristics, which is discussed in (Ball, 2011): Each solution is composed of single timetabled services, which induces an intuitive decomposition, where each partial problem can be optimized by applying the existing model.

Applying the LMNS algorithm, we obtain solutions for instances based on real-world data from the German market in attractive computation times. Example instances where we can determine optimal solutions confirm the high quality of the heuristic solutions we obtain. Indeed, the optimal solution is found for 101 out of 102 instances with known optimal solution. Moreover, we show that the explicit consideration of the dependencies between the different timetabled services often produces insightful new results that differ from approaches which only focus on a single service.

The remainder of this paper is structured as follows: We review the existing literature with respect to integrated and schedule-based network planning as well as the algorithmic approach in Section 2. The new model is presented in Section 3 and the solution approach, which is based on a large multiple neighborhood search (LMNS), in Section 4. Subsequently, we discuss computational performance and selected model outputs in Section 5. We conclude by summarizing our findings and discussing possible next steps for research in schedule-based public transport planning and the integration of planning steps in Section 6.

2. Literature review

This section is divided into three parts covering literature on integrated and schedule-based line planning in Section 2.1, publications relevant from an algorithmic perspective in Section 2.2, and a discussion of the positioning and contribution of this paper in Section 2.3.

2.1. *Integrated and schedule-based line planning*

A comprehensive survey on the line planning step in public transportation was presented by Schöbel (2011). The focus area of our paper is the integration of planning steps, in particular schedule-based approaches and considerations of dynamic demand. These aspects and relevant references are discussed in detail in (Steiner and Irnich, 2018), hence we only present the most recent contributions in this paper.

A line of research focusing on the integration of line planning, timetabling and vehicle scheduling is presented in (Schöbel, 2017) and in earlier papers by the same authors. A recent example of integration with the preceding planning step network design is presented by Canca *et al.* (2017). The presented model decides simultaneously on which nodes and edges to include in the network, on line structure and headways, on public transport mode share and passenger routes, and on train capacities. The determination of the public transport mode share is in fact also an approach to include dynamic demand. In (Abdelghany *et al.*, 2017), the authors present a model to optimize the flight schedule of an airline considering dynamic demand effects due to competition with other airlines. In a bi-level model setup, the scheduling decisions are made on the upper level, while the lower level determines the resulting passenger decisions.

2.2. *Large neighborhood search and variations*

The concept of large neighborhood search (LNS) was introduced by Shaw (1998) and an extensive overview including variations is provided in (Pisinger and Ropke, 2010). The general approach of LNS is based on starting with a feasible solution and then alternatingly applying a *destroy* and a *repair* operator to obtain new solutions. A new solution is accepted if an acceptance criterion is fulfilled. In the event that there are multiple destroy and repair operators, the approach is referred to as a large multiple neighborhood search (LMNS). This variation was first introduced by Pisinger and Ropke (2007). The different operators are selected with a predetermined probability throughout the whole algorithm in an LMNS. Meanwhile, adaptive large neighborhood search (ALNS) algorithms continuously adjust these weightings based on the performance of the operators.

LNS, LMNS, and ALNS have been successfully applied to a wide range of problems. In the public transport context, Canca *et al.* (2017) present an ALNS looking at network design and line planning as mentioned above. Further, Hassannayebi and Zegordi (2017) and Barrena *et al.* (2013) developed ALNS algorithms focusing on the timetabling step while integrating aspects of dynamic demand. The earliest and most frequent applications of LNS algorithms focus on the vehicle routing problem (VRP) and related problems. The ALNS approach was first introduced with an application for the pickup and delivery problem with time windows (PDPTW) by Ropke and Pisinger (2006). In (Masson *et al.*, 2013), an ALNS for the pickup and delivery problem with transfers (PDPT) is presented, while Hintsch and Irnich (2018) solve the clustered vehicle routing problem (CluVRP) with an LMNS.

2.3. *Positioning and contribution of this work*

The contribution of this work is twofold: First, we provide a new schedule-based mixed-integer linear model formulation, which is compatible with dynamic demand considerations. As discussed in the literature review of (Steiner and Irnich, 2018), to our knowledge there are no other papers addressing this combination of scopes. Further, in contrast to that previous paper, the model here enables us to optimize multiple timetabled services simultaneously. Second, we present a problem-specific LMNS solution algorithm capable of solving real-world instances in attractive computation times and with high quality solutions.

3. Integrated and schedule-based optimization model

Before presenting the notation and the mixed-integer linear formulation of the model, we make two general comments on the scope of the model. First, similar to (Steiner and Irnich, 2018), a very detailed representation of demand is given as a model input. Specifically, the demand depends on the pair of stations, the departure time, the trip duration, the trip frequency, and the degree of cannibalization of a trip. As a consequence, the model can be applied after having determined the demand parameters with a separate demand model. These demand models can be based on complex approaches, e.g., machine learning. Therefore, we see it as a favorable setup to separate the demand modeling step from the optimization based on mathematical programming.

Second, there is no differentiation between travel prices for a specific pair of stations and a specific timetabled service. In practice, most operators apply a more sophisticated revenue management with prices varying based on how many tickets have been sold already and how many days are left until the trip. However, we focus on the strategic planning of bus operations, whereas the pricing considerations are only relevant at a later stage in practice. This is again an analogous approach to (Steiner and Irnich, 2018).

In the following, the model formulation is presented in Section 3.1 and potential model extensions are discussed in Section 3.2.

3.1. Model formulation

To build on the model formulation and solution algorithm developed in (Steiner and Irnich, 2018), we keep the notation and modeling approach consistent with this paper. For convenience, all basic terms are defined in Table 1. We have a corridor of potential stations s_i indexed by i, $i \in I = \{1, \dots, n\}$. In this corridor, a set of timetabled services is scheduled by the model. Potential departure times at s_1 are denoted by c_m, where the index m runs in the discrete index set M. We refer to the potential timetabled service starting at station s_1 at the time c_m as the m-th timetabled service or the service m. We assume that every selected timetabled service starts at station s_1 and ends at station s_n, e.g., to allow for efficient vehicle schedules in the next planning step. However, this assumption could be relaxed by slightly adjusting the model we present in this chapter.

Possible start times at stations and duration intervals are modeled using discrete time intervals $T_k = [a_{k-1}, a_k)$ and $D_l = [b_{l-1}, b_l)$, where the indexes k and l run in the discrete index sets K and L respectively. The number of times a trip between a pair of stations s_i and s_j is offered is referred to as trip frequency and denoted by $f \in \mathbb{N}$. Finally, we assign a degree of cannibalization g to each trip between s_i and s_j, where g runs in the discrete index set G. If there are further timetabled services offering the same trip at a similar time, the degree of cannibalization is higher, which has a negative impact on demand for this trip.

To improve legibility, we consistently use indices $m \in M$ for timetabled services, $i \in I$ and $j \in I$ for stations always with $i < j$, $k \in K$ for departure time intervals, $l \in L$ for duration intervals, $f \in \mathbb{N}$ for trip frequencies, and $g \in G$ for degrees of cannibalization. Further, we omit the index sets when summing over the m, i, j, k, l, f, and g and we assume that all index sets $M, I, K, L, \mathbb{N}$, and G are pairwise disjoint.

The model formulation requires the following *input data*:

d_{ijklfg} — demand for a trip between s_i and s_j, which starts in $T_k = [a_{k-1}, a_k)$ with duration in $D_l = [b_{l-1}, b_l)$, is operated f times, and has a degree of cannibalization g;

Term	Description
Corridor	is a sequence $(s_1, s_2, \dots, s_n)$ of stations, from which a subsequence must be selected as stops of the timetabled services.
Timetabled Service	is a run from s_1 to s_n of a bus on a specified subsequence of stations s_i with a specified schedule; the schedule is implicitly given by the departure time c_m at station s_1.
Trip	is a pair of two (selected) stations s_i and s_j (with $i < j$) that are connected either directly or via intermediate stops by a timetabled service; a trip is what customer demand refers to.
Trip frequency	is the count of trips between two stations s_i and s_j in the time period in scope (e.g., one day). Only trips of the operator in scope of the model are considered.
Cannibalization	refers to the negative effect on the demand in case multiple trips between two stations with similar starting times are offered by the operator in scope.
Direct Connection	is a pair of two consecutive stations s_i and s_j without intermediate stop; this is where passengers and bus travel along; direct connections are modeled as basis for operational costs.

Table 1: Definitions of basic terms

- t_{mij} travel time of the m-th timetabled service for a direct connection from s_i to s_j including the stop time at s_j;
- w_{mi} stop time of the m-th timetabled service at station s_i for handling of luggage, boarding, schedule buffer, etc.;
- r_{mij} travel prices (revenues from the operator's perspective) of the trip from s_i to s_j for the m-th timetabled service;
- v_{mij} variable cost for the m-th timetabled service to operate a direct connection from s_i to s_j;
- f_{ml} fixed cost to operate the m-th timetabled service from s_1 to s_n with duration in D_l, this captures the share and period of the day when the bus is dedicated to the service in scope;
- C_m vehicle capacity (number of seats of a bus) for the m-th timetabled service;
- F Maximum number of timetabled services to be operated during the time period in scope (e.g., one day).

All these inputs are non-negative numbers. Although the actual amount of passengers per trip is integer, we do not impose integrality for the d_{ijklfg}, since we are dealing with the strategic/tactical planning stage. Moreover, let M_{mik} and M_{mijl} be sufficiently large numbers (*big M* constants), and let $u \in \mathbb{R}$ be a small time amount (e.g., one minute) that we use to transform $<$ into $\leq$ conditions.

The model formulation comprises four types of decision variables describing the characteristics of timetabled services that are selected by the model. The remaining types of variable are auxiliary indicator variables and are presented below.

- $y_m \in \{0; 1\}$ binary variable indicating the m-th timetabled service starting at station s_1 at time c_m is operated;
- $x_{mi} \in \{0; 1\}$ binary variable to indicate the station s_i is included in the m-th timetabled

service;

$p_{mij} \in \mathbb{R}_{\geq 0}$ continuous variable for the number of passengers for a trip from s_i to s_j in the m-th timetabled service;

$\ell_{mi} \in \mathbb{R}_{\geq 0}$ continuous variable for the duration of m-th timetabled service to reach s_i while considering all chosen intermediary stations. This is a dependent variable, its value can be determined once the variables y_m and x_{mi} are fixed.

The remaining seven types of binary variables display the logical links between the stations, time intervals, trip frequencies, and degrees of cannibalization. They take the value 1 if and only if the selection of timetabled services, the choice of stations, the departure time, the duration, the trip frequency, and the degree of cannibalization are consistent with the indices $m \in M$, $i, j \in I$, $k \in K$, $l \in L$, $f \in \mathbb{N}$, and $g \in G$. All these variables are denoted by z and corresponding index sets. Since $M, I, K, L, \mathbb{N}$, and G are pairwise disjoint, the following definitions are unambiguous:

$z_{mijklfg}$ The m-th timetabled service contains a trip from s_i to s_j, which starts in T_k at station s_i with duration in D_l. The trip from s_i to s_j is operated f times and there is a degree of cannibalization g;

z_{mij} The m-th timetabled service contains a *direct* connection (no intermediary stops) from s_i to s_j;

z_{ml} The m-th timetabled service is operated with total duration in D_l to reach the destination s_n;

z_{mik} The m-th timetabled service contains a trip which starts at s_i in T_k;

z_{mijl} The duration for the trip from s_i to s_j of the m-th timetabled service is in D_l;

z_{ijf} There are exactly f different timetabled services offering trips from s_i to s_j;

z_{mijg} The trip from s_i to s_j contained in the m-th timetabled service has degree of cannibalization g.

To clarify the problem setting and notation introduced above, we provide a small example before presenting the mixed-integer linear model formulation.

Example. Consider a corridor (s_1, s_2, s_3) with three stations and the m-th timetabled service to start at time $c_m = 10(m-1)$, if it is selected. To explain the decision variables in more detail, we base our example on sample solutions and discuss the impact on the variables. For the sake of convenience, we use commas between the indices in this example. We assume that the 1st service starting at $c_1 = 0$ as well as stations s_1 and s_3 are selected, i.e., $y_1 = x_{1,1} = 1 - x_{1,2} = x_{1,3} = 1$. Further, we assume the 2nd and 5th service and all their stations are selected, i.e., $y_2 = y_5 = x_{2,1} = x_{2,2} = x_{2,3} = x_{5,1} = x_{5,2} = x_{5,3} = 1$. For the variables z_{mij} representing direct connections, this implies $z_{1,1,3} = z_{2,1,2} = z_{2,2,3} = z_{5,1,2} = z_{5,2,3} = 1$.

The following assumptions on input data and cannibalization dynamics are made for this example: We assume travel times $t_{mij} = 3(j-i)+1$ for all m and $i < j$ as well as stop times $w_{mi} = 1$ for all m, i (note the t_{mij} have been defined to include the stop time at s_j). Start times are discretized by $T_k = [k-1, k)$ and durations by $D_l = [l-1, l)$. For a trip between stations s_i and s_j starting at time t, the degree of cannibalization g is determined as follows: Among the timetabled services including a trip from s_i to s_j, we select the one with starting time t^* at s_i, such that $|t^* - t|$ is minimal, i.e., the trip with the closest possible starting time. The degree of cannibalization is given by $g = 20 - |t^* - t|$ in case $|t^* - t| < 20$ and $g = 0$ otherwise. In the event that there is no other trip from s_i to s_j, the degree of cannibalization is 0 as well. Hence, the maximum possible degree of cannibalization is 20 in case two trips start at the exact same time. Demand would in general decrease with an increasing degree of cannibalization.

With the services and stations selected as described above, a total of seven trips are included in the three selected timetabled services. Table 2 provides details for each trip and displays, which of the $z_{mijklfg}$ would take the value 1 in a solution of our model based on the assumptions made.

timetabled			start	end	$z_{mijklfg} = 1$ for			
service m	station i	station j	time	time	k	l	f	g
1	1	3	0	6	1	7	3	10
2	1	2	10	13	11	4	2	0
2	1	3	10	17	11	8	3	10
2	2	3	14	17	15	4	2	0
5	1	2	40	43	41	4	2	0
5	1	3	40	47	41	8	3	0
5	2	3	44	47	45	4	2	0

Table 2: Trip characteristics for small example

The resulting demand for the trip between s_1 and s_3 offered by the 1st timetabled service is $d_{1,3,1,7,3,10}$. Assuming the 5th service had not been selected, the demand would change to $d_{1,3,1,7,2,10}$, as we still observe the cannibalization effect between the first and second timetabled service, however only two trips between the stations s_i and s_j are still offered. If we further assume that also the 2nd service had not been selected, the demand would be $d_{1,3,1,7,1,0}$. The trip frequency would reduce to 1 and there would clearly be no cannibalization effect with other services, as there is only one service remaining.

After selecting timetabled services and their stations as well as computing durations, the number of customers to assign to the trips must be determined. The p_{mij} variables are constrained by the respective demand parameters d_{ijklfg} and by the vehicle capacity. As an example, for the 2nd timetabled service, the choice is constrained by $p_{2,1,2} \leq d_{1,2,11,4,2,0}$, $p_{2,1,3} \leq d_{1,3,11,8,3,10}$, as well as $p_{2,2,3} \leq d_{2,3,15,4,2,0}$. Further, the restricted capacity yields $p_{2,1,2} + p_{2,1,3} \leq C_2$ and $p_{2,1,3} + p_{2,2,3} \leq C_2$, which induces a multi-commodity network-flow optimization problem. □

Mixed-integer linear formulation. We now step systematically through the model formulation (1)–(11b). The overall structure is similar to the model from (Steiner and Irnich, 2018). The main differences are the additional indices m, f, and g as well as the constraints on trip frequencies and cannibalization.

The objective (1) is to maximize profit, thus, to maximize revenues minus fixed and variable costs of all selected timetabled services. Fixed costs depend on the departure times and the overall durations of the selected timetabled services, and variable costs depend on the selected stations within the timetabled services:

$$\max \sum_m \Big(\sum_{i<j} r_{mij} p_{mij} - \sum_l \mathrm{f}_{ml} z_{ml} - \sum_{i<j} \mathrm{v}_{mij} z_{mij} \Big) \tag{1}$$

subject to

$$\sum_{klfg} z_{mijklfg} \leq x_{mi}, \quad \forall i < j, \forall m \tag{2a}$$

$$\sum_{klfg} z_{mijklfg} \leq x_{mj}, \quad \forall i < j, \forall m \tag{2b}$$

$$\sum_{lfg} z_{mijklfg} \leq z_{mik}, \quad \forall i < j, \forall k, m \tag{2c}$$

$$\sum_{kfg} z_{mijklfg} \leq z_{mijl}, \quad \forall i < j, \forall l, m \tag{2d}$$

$$\sum_{klg} z_{mijklfg} \leq z_{ijf}, \quad \forall i < j, \forall f, m \tag{2e}$$

$$\sum_{klf} z_{mijklfg} \leq z_{mijg}, \quad \forall i < j, \forall g, m \tag{2f}$$

Passengers may only enter or exit a bus at those stations s_i and s_j, which have been included (2a)–(2b), in the departure interval T_k at s_i that actually contains the departure time of the trip (2c), and the duration needs to be in the correct duration interval D_l (2d). Further, the $z_{mijklfg}$ can only take the value 1 if the corresponding z_{ijf} and z_{mijg} are set to 1 as well (2e)–(2f).

$$p_{mij} \leq \sum_{klfg} d_{ijklfg} z_{mijklfg}, \quad \forall i < j, \forall m \tag{3a}$$

$$\sum_{i' \leq i, j' > i} p_{mi'j'} \leq C_m, \quad \forall i < n, m \tag{3b}$$

The number of passengers per trip is constrained by the demand (3a) and must not exceed the capacity of the bus on each connection (3b).

$$\sum_{j>1} z_{m1j} = \sum_{i<n} z_{min} = y_m, \quad \forall m \tag{4a}$$

$$\sum_{j<i} z_{mji} = \sum_{j>i} z_{mij}, \quad \forall 1 < i < n, \forall m \tag{4b}$$

$$\sum_{j>i} z_{mij} = x_{mi}, \quad \forall 1 < i < n, \forall m \tag{4c}$$

$$z_{ml} + 1 \geq y_m + z_{m1nl}, \quad \forall l, m \tag{4d}$$

The flow conditions (4a)–(4c) ensure that the z_{mij} only take the value 1 if the m-th timetabled service and both stations are included, and there are no intermediate stations between them. The incorporation of fixed costs f_{ml} results from $z_{ml} = 1$, which is ensured by (4d) if the m-th timetabled service has a total duration in D_l.

$$\sum_{m} y_m \leq F \tag{5}$$

$$\ell_{mi} = \sum_{i_1 < j_1 \leq i} t_{mi_1j_1} z_{mi_1j_1}, \quad \forall i, m \tag{6}$$

$$x_{m1} = x_{mn} = y_m, \quad \forall m \tag{7}$$

At most F timetabled services can be selected (5) and the duration to reach station s_i results from the selected connections to reach s_i (6). As discussed above, we request the first and the last station to be included in each selected timetabled service (7).

$$\sum_k z_{mik} = x_{mi}, \ \ \forall i < n, m \tag{8a}$$

$$\sum_k z_{mik} \leq y_m, \ \ \forall i < n, m \tag{8b}$$

$$c_m + \ell_{mi} \leq a_k + (1 - z_{mik})M_{mik} - u, \ \ \forall i < n, k, m \tag{8c}$$

$$c_m + \ell_{mi} \geq a_{k-1} z_{mik}, \ \ \forall i < n, k, m \tag{8d}$$

Variable z_{mik} can only take the value 1 if the m-th timetabled service is selected and services station s_i (8a)–(8b). Consistency with the travel and departure times results from (8c) and (8d), which ensure z_{mik} can only take the value 1 if the starting time at s_i (which can be written as $c_m + \ell_{mi}$) is smaller than a_k and greater than or equal to a_{k-1}.

$$\sum_l z_{mijl} \geq x_{mi} + x_{mj} - 1, \ \ \forall i < j, m \tag{9a}$$

$$\sum_l z_{mijl} \leq x_{mi}, \ \ \forall i < j, m \tag{9b}$$

$$\sum_l z_{mijl} \leq x_{mj}, \ \ \forall i < j, m \tag{9c}$$

$$\ell_{mj} - \ell_{mi} - w_{mj} \leq b_l + (1 - z_{mijl})M_{mijl} - u, \ \ \forall i < j, \forall l, m \tag{9d}$$

$$\ell_{mj} - \ell_{mi} \geq (b_{l-1} + w_{mj}) z_{mijl}, \ \ \forall i < j, \forall l, m \tag{9e}$$

Likewise, the variable z_{mijl} can only take the value 1 if and only if both stations s_i and s_j are included (9a)–(9c). Further, (9d) and (9e) enforce the duration interval to be chosen consistently with the actual travel time from s_i to s_j (which can be written as $\ell_{mj} - \ell_{mi} - w_{mj}$).

$$\sum_f z_{ijf} \leq 1, \ \ \forall i < j \tag{10a}$$

$$\sum_{ml} z_{mijl} = \sum_f f z_{ijf}, \ \ \forall i < j \tag{10b}$$

$$\sum_l z_{mijl} = \sum_g z_{mijg}, \ \ \forall i < j, m \tag{10c}$$

For a pair of stations s_i and s_j, at most one variable z_{ijf} can take the value 1 (10a) and this is only possible if the trip frequency takes indeed the value f (10b). Additionally, for each selected timetabled service and pair of stations s_i and s_j, one degree of cannibalization needs to be selected, this is enforced by (10c).

To avoid another binary variable indicating a timetabled service includes stations s_i and s_j (not necessarily as a direct connection), the left hand sides of (10b) and (10c) use the sum over the variables z_{mijl}. Indeed, exactly one of them takes the value 1 by (9a)–(9c) in case the m-th timetabled service includes both stations s_i and s_j.

$$z_{mik} + x_{mj} + z_{m'ik} + x_{m'j} \leq 3 + z_{mijg_2}, \quad \forall i < j, \forall k, m, m', m \neq m' \tag{11a}$$

$$z_{mik} + x_{mj} + z_{m'i(k-1)} + z_{m'i(k+1)} + x_{m'j} \leq 3 + \sum_{g \in \{g_1, g_2\}} z_{mijg}, \quad \forall i < j, \forall k, m, m', m \neq m' \tag{11b}$$

For each selected timetabled service m and pair of selected stations s_i and s_j, the degree of cannibalization is controlled by (11a) and (11b). For this paper, we have chosen $G = \{g_0, g_1, g_2\}$, with g_2 indicating a *high degree of cannibalization*, g_1 a *medium degree of cannibalization*, and g_0 that there is *no cannibalization* at all. The high degree of cannibalization g_2 is enforced if there are two distinct timetabled services m and m', which both contain a trip from station s_i to s_j starting in the same interval T_k. In this case, all four terms of the left hand side of (11a) take the value 1 and thus force z_{mijg_2} to the value 1 as well. Similarly, in case these two trips from s_i to s_j do not start in the same time interval T_k, but in chronologically neighboring intervals (e.g., T_{k-1} and T_k), a minimum degree of cannibalization g_1 is assumed. If so, the left hand side of (11b) takes the value 4 (since $z_{m'i(k-1)}$ and $z_{m'i(k+1)}$ cannot take the value 1 simultaneously), which forces at least the degree of cannibalization g_1.

Note that there is no unique or mandatory logic to model cannibalization and the above formulation is just one possibility to capture it. If a heuristic solution algorithm is applied, even non-linear approaches can be considered in the event that these are best suited to capture the results of the demand modeling step. Assuming the possible degrees of cannibalization $g \in G$ can be ordered, the formulation (11a)–(11b) can be generalized to a set of constraints, where each constraint enforces at least a certain degree of cannibalization g_γ. Here, the left hand side includes the variables that indicate a cannibalization impact of degree g_γ on the trip of service m from station s_i to s_j starting in T_k. Further, the right hand side comprises an integer parameter (in our case its value is 3 in all cases) such that one of the variables z_{mijg} for $g \geq g_\gamma$ needs to take the value 1 if the left hand side takes its maximum value.

3.2. Model extensions

As formulated above, the model (1)–(11b) can select stations for two distinct timetabled services m and m' independently. This strategy makes sense from a customer and from an operator perspective: Passengers have access to a wider range of trips and these are designed and scheduled to fit well with the demand structure. Given the popularity of online journey planners, passengers do not need rules such as "line l always stops at station s" any more. Yet, operators can maximize their profit without including additional constraints, which could deteriorate the solution quality.

However, it could be desired from a regulatory or convenience perspective to operate timetabled services on lines with identical or at least very similar sequences of stations. In the following, we discuss how the presented model can be adjusted to incorporate these requirements. In the event that every selected timetabled service should contain exactly the same stations, one additional type of variables x_i can be introduced, which indicates that the station s_i is included in all selected timetabled services. Additional constraints

$$x_{mi} \leq x_i, \ \forall m, i \quad \text{and} \quad x_i + y_m \leq 1 + x_{mi}, \ \forall m, i$$

enforce this logic. Starting with the above requirement of identical stations and assuming each selected timetabled service can contain one additional selected station (which is not selected by all services, i.e., the corresponding x_i takes the value 0), a similar approach can be taken with the same variable x_i. Now, the constraints

$$\sum_i x_{mi} \leq \sum_i x_i + 1, \ \forall m \quad \text{and} \quad x_i + y_m \leq 1 + x_{mi}, \ \forall m, i$$

can be added to realize the requirement. We analyze the impact of including such additional requirements in Section 5.5.

Finally, the two extensions for back-and-forth services and aspects around driver scheduling, which are discussed in (Steiner and Irnich, 2018), can analogously be applied to the model (1)–(11b).

4. LMNS-based solution algorithm

The objective of this work is to solve real-world instances based on the model (1)–(11b). Given the complexity of the model and the size of real-world instances, a heuristic approach seems most promising. We decided for a large multiple neighborhood search (LMNS) for three key reasons. First, approaches based on LNS have been applied successfully to a range of similar real-world problems as discussed in Section 2.2. Second, we see an intuitive way to define neighborhood structures when given a solution of the model (1)–(11b): Larger steps within the solution space to avoid being trapped in local optima can be performed by adding, deleting or shifting entire timetabled services from the current solution. Local exploration is possible by adjusting the timetabled services that are already present in the current solution. Third, the structure of solutions suggests the application of multiple operators. A combination of adding, deleting, and shifting entire timetabled services as well as selected stations seems more promising than deciding for just one operator.

We have opted against an adaptive layer for the operator selection: Since the problem we study has not been studied before in this form, we believe it is beneficial to better understand the benefit of each operator without the additional influence and variety of parameters of the adaptive layer. Further, given the different computational complexity of the operators we use, the adaptation logic would need to include the time spent by each operator, which creates challenges for the replicability of results. Finally, pre-tests including an adaptive layer did not show a consistent picture of certain operators being powerful only early in the algorithm and not in later iterations or vice-versa.

The set of operators we apply is introduced in Section 4.1 and different operator application strategies are discussed in Section 4.2. The overall LMNS algorithm is presented in Section 4.3.

4.1. LMNS operators

Typically, LNS operators can be classified into *destroy* and *repair operators*. Here, a destroy operator deletes or removes certain parts of a solution, which gives a partial solution. This partial solution is then transformed again into a feasible solution by the repair operator. In the context of vehicle routing problems (VRP) and related problems, the destroy operator often removes entire vehicle tours or specific customers from within a tour. The repair operator then inserts the removed customers based on either random, heuristic or optimization-based approaches.

In our case, the situation differs from the VRP context: Indeed, any given set of values for the y_m and x_{mi} yields a solution of the model (1)–(11b) after solving the multi-commodity network-flow problem to determine the optimal passenger flows. Therefore, we do not have the differentiation between destroy and repair operators.

The operators we apply in the LMNS solution algorithm can be clustered along three main dimensions: First, the operator moves are of different *types*: operators either *add, delete* or *shift* parts of the solution, i.e., entire timetabled services or stations within a selected timetabled service.

Second, certain operators mainly serve the *purpose* to *intensify* the search to find local optima, whereas the remaining operators *diversify* the current solution. The intensification operators retain the selected timetabled services and only add, delete or shift selected stations. Meanwhile, the diversification operators modify the given solution by adding, deleting or shifting entire timetabled services.

Third, the *degree of randomness* varies from operators based on *random* modification of the current solution to *best* operators that perform modifications based on the best possible impact of the operator application on the objective function. Still, a degree of randomization similar to (Ropke and Pisinger, 2006, p. 459) is included in the *best* operators to increase the diversification of the overall LMNS algorithm. For the *best* operators, we differentiate between *heuristic best* and *optimized best* operators. The aim of the heuristic operators is to combine the advantages of forward-looking and fast modifications. In particular, these operators avoid to apply any optimization model. Hence, the effect on the objective function is pre-estimated based on information that can be calculated easily without calling the multi-commodity network-flow model for determining the precise objective value. Meanwhile, the optimized operators determine the best possible modifications of the current solution.

Finally, we include one more operator that is based on the optimization model (1)-(C2) for single timetabled services presented in (Steiner and Irnich, 2018). As this operator comprises a complex optimization algorithm, heuristics based on this operator can be categorized as *matheuristics*. Altogether, we have a list of 19 operators displayed in Table 3. Based on these operators, we present different setups and operator application strategies in Section 4.2 and analyze the performance of the resulting heuristics in Section 5.

purpose	type	degree of randomness			
		random	heuristic best	optimized best	matheuristic operator
Intensification (adjust stations)	add	1.1.	1.2.	1.3.	7.
	delete	2.1.	2.2.	2.3.	
	shift	3.1.	3.2.	3.3.	
Diversification (adjust services)	add	4.1.	4.2.	4.3.†	
	delete	5.1.	5.2.	5.3.	
	shift	6.1.	6.2.	6.3.	

Table 3: Overview of LMNS operators; †not included in LMNS due to long computation times

Each operator *op* has an extent of modification, which we denote by $S \in \mathbb{N}$. This is the number of stations or timetabled services, which are added, deleted or shifted by the operator. The selection of S is performed at random before the application of an operator in a way that ensures it is indeed possible to add, delete or shift S stations or timetabled services. We analyze the impact of varying S in Section 5. For $S > 1$, the S modifications are realized sequentially. We use the term *iteration* for a single step and denote the specific iteration we describe by $iter_{op}$.

We now describe each of the 19 operators in more detail.

Intensification operators.

1. *Add station operators* add $S \in \mathbb{N}$ stations within a preselected timetabled service m.

   1.1. *Add random stations* adds S stations at random in service m.

   1.2. *Heuristic add best stations* adds S stations in service m based on an estimation of their contribution to the objective function. For every station s_{i^*} to be added directly between stations s_i and s_j, the contribution is estimated by

   $$\mathrm{con}^1_{m,i^*} = \sum_{\substack{i'<i^* \\ i' \in I_m}} r_{mi'i^*} d_{i'i^*klfg} + \sum_{\substack{i^*<j' \\ j' \in I_m}} r_{mi^*j'} d_{i^*j'klfg} + \mathrm{f}_{ml_1} - \mathrm{f}_{ml_2} + \mathrm{v}_{mij} - \mathrm{v}_{mii^*} - \mathrm{v}_{mi^*j}.$$

   Here, I_m is the set of selected stations in the service m and the respective indices for k, l, f, and g for the demand parameter d are determined assuming the service includes station s_{i^*}. Further, D_{l_1} denotes the duration interval of the total travel time of the service before, and D_{l_2} after adding station s_{i^*}. The demand parameter d is used instead of the variable p, which appears in the objective function. This is done to avoid having to solve the multi-commodity network-flow problem within the heuristic operator.

   The calculated contributions con^1_{m,i^*} are ranked in descending order and the station at position $\lfloor \alpha^\rho \cdot (n - n_m) \rfloor$ is added. Here, $\alpha \in [0, 1)$ denotes a uniformly distributed random variable, $\rho \in \mathbb{N}$ with $\rho \geq 1$ controls the degree of randomization (as introduced in (Ropke and Pisinger, 2006, p. 459)), and n_m is the number of stations selected in the m-th timetabled service of the current solution. For $S > 1$, a new value for α is randomly selected and the contributions are updated after each iteration $iter_{op}$.

   1.3. *Optimized add best stations* adds S stations in service m based on their exact contribution to the objective function. Calculations are performed updating the demand parameters and by solving the multi-commodity network-flow model for every station from service m that could be added. An analogous approach to the *heuristic add best stations* operator is followed for ranking and randomized adding of a station.

2. *Delete station operators* delete $S \in \mathbb{N}$ stations within a preselected timetabled service m.

   2.1. *Delete random stations* deletes S stations at random from service m.

   2.2. *Heuristic delete best stations* deletes S stations from service m based on an estimation of their contribution to the objective function. For every station s_{i^*} to be deleted directly between stations s_i and s_j, the contribution is estimated by

   $$\mathrm{con}^2_{m,i^*} = -\sum_{i'<i^*} r_{mi'i^*} p_{mi'i^*} - \sum_{i^*<j'} r_{mi^*j'} p_{mi^*j'} + \mathrm{f}_{ml_1} - \mathrm{f}_{ml_2} - \mathrm{v}_{mij} + \mathrm{v}_{mii^*} + \mathrm{v}_{mi^*j}.$$

   The values of the p-variables are based on the accepted solution of the LMNS algorithm. This time, D_{l_1} denotes the duration interval of the total travel time of the service before, and D_{l_2} after deleting station s_{i^*}. Note that this is still a heuristic approach, because the multi-commodity network-flow problems would need to be solved for an exact contribution. Indeed, the demand values for service m change due to the modified departure and travel times. Further, for the services $m' \neq m$ the demand is affected as well due to the effect of the deleted station on trip frequencies and cannibalization.

The calculated contributions con^2_{m,i^*} are ranked in descending order and the station at position $\lfloor \alpha^\rho \cdot n_m \rfloor$ is deleted, where n_m is the number of selected stations in the m-th service in the current solution. Recall that we request the stations s_1 and s_n to be included in every timetabled service. Therefore, we do not consider the option of deleting these stations and use in fact $n_m - 2$. This requirement is reflected in an analogous way in the other operators and is not explicitly pointed out in the following. Only the cost part of the solution is updated after each iteration $iter_{op}$, because an update of the revenue contribution would require solving the multi-commodity network-flow model.

2.3. *Optimized delete best stations* deletes S stations from service m based on the exact objective value after deleting the stations. As before, calculations are performed by solving the multi-commodity network-flow model and the ranking of contributions as well as the randomized selection of the station to be deleted are analogous to the *heuristic delete best stations* operator.

3. *Shift station operators* shift $S \in \mathbb{N}$ stations within a preselected timetabled service m (i.e., a selected station is deleted and a non-selected station is added instead). To increase the level of diversification, we make the following restrictions if a shift from s_i to s_j has already been performed in an earlier iteration: Station s_j needs to stay selected and station s_i can not be selected again.

3.1. *Shift random stations* shifts S stations at random.

3.2. *Heuristic shift best stations* shifts S stations in service m based on an estimation of their contribution to the objective function. For every combination of a selected station s_{i^*} directly between stations s_i and s_j and a non-selected station s_{j^*} directly between stations s'_i and s'_j within service m, the contribution of deleting s_{i^*} and adding s_{j^*} is estimated by

$$\text{con}^3_{m,i^*,j^*} = \text{con}^2_{m,i^*} + \text{con}^1_{m\setminus\{i^*\},j^*}.$$

First, the impact of deleting s_{i^*} is estimated analogously to the *heuristic delete best station* operator. Subsequently, station s_{j^*} is added to the service denoted by $m \setminus \{i^*\}$, i.e., to the m-th timetabled service without station s_{i^*}. Here, the impact is estimated as before for the *heuristic add best station* operator.

The calculated contributions con^3_{m,i^*,j^*} of the combinations (i^*, j^*) are ranked in descending order and, with the notation from above, the shift for the combination at position $\lfloor \alpha^\rho \cdot n_{\text{shift}} \rfloor$ is performed. The number of possible combinations of stations is denoted by n_{shift}, which is given by $(n_m - iter_{op} + 1) \cdot (n - n_m - iter_{op} + 1)$ in iteration $iter_{op}$. All contribution aspects are updated after each iteration $iter_{op}$, except for the lost revenue of not servicing a shifted station any more.

3.3. *Optimized shift best stations* shifts S stations in service m based on their exact contribution to the objective function. For every allowed combination of a station that is included and a station that is not included, the impact of the potential shift on the objective value is calculated with the multi-commodity network-flow model. Ranking and randomized selection of the shift to perform are analogous to the *heuristic shift best stations* operator.

Diversification operators.

4. *Add service operators* add $S \in \mathbb{N}$ timetabled services.

   4.1. *Add random services* adds S timetabled services at random. Within the added timetabled services, the stations to be included in addition to s_1 and s_n are also selected randomly.

   4.2. *Heuristic add best services* adds S timetabled services based on an estimation of their potential contribution to the objective function. For every timetabled service m that is not included in the current solution, the contribution is estimated as follows assuming every station is included in the added service:

$$\mathrm{con}_m^4 = \sum_{i<j} r_{mij} d_{ijklfg} - \mathrm{f}_{ml_1} - \sum_{(i,j)\in I_m^*} \mathrm{v}_{mij}.$$

   The first term is the revenue potential and the respective indices for k, l, f, and g for the demand parameter d are determined assuming all stations are included. The second and third term represent costs, where D_{l_1} denotes the duration interval of the total travel time of the service including all stations, and I_m^* is the set of indices for direct connections between neighboring stations s_i and s_{i+1}. In this approach, the capacity constraint is neglected to avoid having to solve the multi-commodity network-flow problem.
   The calculated contributions con_m^4 are ranked in descending order and the service at position $\lfloor \alpha^\rho \cdot (|M| - n_F) \rfloor$ is added, where n_F is the number of timetabled services included in the current solution. The revenue potential is updated after each iteration $iter_{op}$.
   Typically, timetabled services in good solutions do not include all stations, therefore we only include the stations s_{i^*} with an above average revenue potential

$$\mathrm{rev}_{m,i^*}^4 = \sum_{i<i^*} r_{mii^*} d_{ii^*klfg} + \sum_{i^*<j} r_{mi^*j} d_{i^*jklfg}.$$

   4.3. *Optimized add best services* adds S timetabled services based on their exact contribution to the objective function. To avoid having to solve multiple multi-commodity network-flow problems for every possible added timetabled service and every possible constellation of included stations, we apply the model (1)–(11b) with the following adaptations: We fix the variables of the timetabled services that are included in the current solution and require S additional timetabled services to be selected by introducing an additional constraint $\sum_m y_m = n_F + S$. However, pretests have confirmed the intuitive assumption that this operator does not solve to optimality even for smaller instances due to the size of the model (1)–(11b) for real-world setups. It is therefore not included in the LMNS in the remainder of this paper.

5. *Delete service operators* delete $S \in \mathbb{N}$ timetabled services.

   5.1. *Delete random services* deletes S timetabled services at random.

   5.2. *Heuristic delete best services* deletes S timetabled services based on an estimation of the change in objective value in case these services are deleted. For a selected timetabled service m, the contribution is estimated by

$$\mathrm{con}_m^5 = -\sum_{i<j} r_{mij} p_{mij} + \mathrm{f}_{ml_2} + \sum_{(i,j)\in I_m^*} \mathrm{v}_{mij}.$$

The values of the p-variables are based on the accepted solution of the LMNS algorithm. In the cost terms, D_{l_2} denotes the duration interval of the total travel time of the m-th service, and I^*_m is the set of indices for direct connections in the m-th service. Note that this is still only an approximation of the actual objective value because the effects of deleting the m-th service on the trip frequencies between two stations s_i and s_j as well as on the cannibalization are not considered explicitly. These aspects would change the demand parameters d in the services $m' \neq m$.

Subsequently, the calculated contributions con^5_m are ranked in descending order and the service at position $\lfloor \alpha^\rho \cdot n_F \rfloor$ is deleted. The contributions are not updated after an iteration $iter_{op}$ to avoid having to solve the multi-commodity network-flow model.

5.3. *Optimized delete best services* deletes S timetabled services based on the exact objective value in case these services are deleted. This includes all effects on trip frequencies and cannibalization and is calculated by solving the corresponding multi-commodity network-flow problems for the remaining timetabled services. Analogously to the *heuristic delete best services* operator, the services are ranked by decreasing contribution and a randomized selection of the service to be deleted is performed.

6. *Shift service operators* shift $S \in \mathbb{N}$ timetabled services. The selected stations of the shifted services do not change. To increase the level of diversification, a service that has already been shifted in an earlier iteration is not shifted again.

6.1. *Shift random services* shifts S timetabled services at random.

6.2. *Heuristic shift best services* shifts S timetabled services based on an estimation of the impact on the objective function. For every combination of a selected timetabled service m and a non-selected potential service m', the contribution of performing this shift is estimated by

$$\text{con}^6_{m,m'} = \text{con}^5_m + \sum_{(i,j)\in I'_m} r_{m'ij} d_{ijklfg} - \mathrm{f}_{m'l_1} - \sum_{(i,j)\in I^*_m} \mathrm{v}_{m'ij}.$$

This represents the concatenation of deleting service m and adding in service m' with the same selected stations s_i that were selected in the m-th service. As before, the set I^*_m denotes pairs of indices for direct connections in the m-th service. Further, I'_m contains indices for all pairs of stations s_i and s_j, where both stations are selected in the m-th service.

The contributions $\text{con}^6_{m,m'}$ of the combinations (m, m') are ranked in descending order and we choose the combination at position $\lfloor \alpha^\rho \cdot n_{\text{shift}} \rfloor$. Here, n_{shift} is the number of possible combinations, which is given by $(n_F - iter_{op} + 1) \cdot (|M| - n_F)$. All contribution aspects are updated after each iteration $iter_{op}$, except for the lost revenue of not offering a timetabled service any more.

6.3. *Optimized shift best services* shifts S timetabled services based on their exact contribution to the objective function. For every combination of a selected timetabled service and a non-selected potential departure time we calculate the impact of shifting the timetabled service to the new departure time by solving the resulting multi-commodity network-flow problem. Note that the multi-commodity network-flow problem indeed needs to be solved even for the unchanged timetabled services, as the shifting of a service affects cannibalization and thus the demand values d. Ranking and randomized selection of the shift to perform are analogous to the *heuristic shift best services* operator.

Operator based on the existing optimization model for single timetabled services. Finally, we introduce an operator that is based on the exact optimization model (1)-(C2) from (Steiner and Irnich, 2018). Significantly shorter calculation times compared to the integrated model (1)–(11b) due to the smaller model and the efficient branch-and-cut solution algorithm motivate matheuristics based on this operator.

7. *Replace existing service by an optimized service* replaces an existing timetabled service by a new service based on the model (1)-(C2) from (Steiner and Irnich, 2018). This operator combines deleting an existing service, adding a new service, and selecting the included stations of the new service based on an optimization model. Thus, it can be interpreted as a combination of a *diversification* and an *intensification* operator. Different variations of operator 7 are possible regarding which service gets replaced and which starting times of the new service are allowed:

$7^{*,*}$	The selection of the service to be replaced is made within the operator based on a specified logic. Also, the service to be added is determined by the optimization algorithm inside the operator;
$7^{-,*}$	No service is replaced, i.e., the operator only adds the best possible service;
$7^{m_1,*}$	The service m_1 to be replaced is determined outside of the operator. After deleting this service, the operator adds the best possible service, which could also be service m_1 again;
$7^{m_1,m_2}$	The service m_1 to be replaced and the service m_2 to be added are determined outside of the operator.

Note that the model (1)-(C2) does not include the dynamic demand effects with respect to trip frequencies and cannibalization explicitly. However, the parameters d_{ijkl} for the timetabled service to be added can be pre-calculated to reflect the trip frequencies and cannibalization of the current solution.

4.2. LMNS operator application strategies

There are ample possibilities to combine the presented operators into LMNS solution algorithms. In this section, we present three main dimensions for clustering these possibilities and three variations of matheuristics based purely on the operator 7. First, the most fundamental decision for each operator *op* is the respective selection *probability* π_{op} in an iteration of the LMNS. In particular, the probabilities determine whether *op* is included at all in the algorithm, i.e., $\pi_{op} > 0$.

Second, we differentiate between two approaches for the interplay between intensification and diversification operators. Either these can be selected equally in each iteration based on their probabilities, or a *sequential approach* can be taken. In the latter case, each iteration starts with the application of a diversification operator and subsequently applies an intensification operator to the solution obtained after the diversification step. The effect of the intensification operator is only considered if it leads to an improvement in the objective value compared to the solution after the diversification. In the non-sequential case, the probabilities of all operators add up to 100 %, whereas in the sequential case the probabilities of both diversification and intensification operators add up to 100 %. The sequential approach is taken if a binary parameter *sequential_ approach* is set to *true*.

Third, a *final re-optimization* can be conducted based on operator $7^{m,m}$ for every timetabled service m that is included in the best solution. The adjusted solution is accepted if it is a new best solution. This introduces a tradeoff between the additional computation time and the solution quality. This strategy is applied if a binary parameter *final_ reopt* is set to *true*.

Finally, we introduce three variations a, b, and c of a matheuristic based on operator 7. In these algorithms, the total number of timetabled services to be selected is fixed a priori. Starting from the degenerate setup with no service and no stations selected, a timetabled service is added with operator $7^{-,*}$ until the specified number of services is selected. From this point onwards, each iteration replaces one of the selected services by another service, which is determined by a variation of operator $7^{*,*}$. When selecting the service to be replaced within the operator, the services are ranked based on how recently they have been added to the solution. The most recent service is at the bottom of the list and the service to be replaced is selected in a randomized way based on a random variable α and the exponent ρ as before.

For option a, no additional variations are introduced and the specified procedure is followed in each iteration. Option b is similar with the only difference that within the operator $7^{*,*}$ it is not allowed to add again the service that has just been removed. This is realized by fixing the respective variable y_m to the value 0. Option c is again based on option a and uses the sequential approach, i.e., an additional intensification operator is applied at the end of each iteration.

4.3. Overall LMNS algorithm

The pseudo-code of the overall LMNS algorithm is presented in Algorithm 1. In the following, we denote a solution of the model (1)–(11b) by x. I.e., x comprises an array of values for the variables $(y_m, x_{mi}, p_{mij}, \ell_{mi}, z_{mijklfg}, z_{mij}, z_{ml}, z_{mik}, z_{mijl}, z_{ijf}, z_{mijg})$. In fact, it is sufficient to know the values for the y_m, x_{mi}, and p_{mij}, since the values of all other dependent variables can be uniquely determined once these values are given. The objective value of a solution based on (1) is denoted by $obj(x)$. An intensification operator op_{int} which modifies S stations in the preselected timetabled service m and uses the exponent ρ in the randomization is denoted by $\text{op}_{\text{int}}^{S,m,\rho}$. Likewise, a diversification operator op_{div} which modifies S services and uses the exponent ρ is denoted by $\text{op}_{\text{div}}^{S,\rho}$.

In Step 1, an initial solution x is generated. This step is easy in our case, since every setup of selected timetabled services and included stations induces a feasible solution after solving the multi-commodity network-flow model to determine passenger flows. Except for the special cases of the matheuristics described above, we generate initial solutions by applying the operator *add random services* F times starting with the degenerate setup with no timetabled service and no stations selected.

The main loop of the LMNS comprises Steps 2– 27 and is repeated It_{LMNS} times. In Steps 3 and 4, the best solution is updated if required. The calculation of the objective value $obj(x)$ requires solving the multi-commodity network-flow model to determine passenger volumes.

Subsequently, Steps 5 and 6 update the accepted solution if an acceptance criterion AccCr is fulfilled. In our LMNS, we apply the record-to-record acceptance criterion, which accepts solutions x with $obj(x) > (1-\epsilon) \cdot obj(x^{best})$, where ϵ decreases linearly with every iteration from a starting value ϵ_0 to 0. We decided for this criterion as it allows for simpler parameter calibration than a more complex approach (e.g., simulated annealing) and provides similar solution quality as analyzed in (Santini *et al.*, 2018). Steps 7 and 8 set x to the last accepted solution in case the acceptance criterion is not fulfilled.

Depending on the binary parameter *sequential_ approach*, Step 9 decides which operator selection strategy is chosen. For the sequential approach, the diversification operator op_{div} to be applied to the current solution x is selected in Step 10. The selection is performed randomly based on the operator probabilities π_{op}. Subsequently, the number S of services to be adjusted is selected in Step 11. A random selection with an equal distribution for all feasible values of S is performed. The function $\mathcal{S}$ determines the upper bound for S and ensures $S \leq S_{\max}$ for a global parameter $S_{\max}$. In addition, the solution after the application of op_{div} still needs to contain a minimum of one service and a maximum of F services. In the extreme case of $\mathcal{S}(x, \text{op}_{\text{div}}, S_{\max}) = 0$ (e.g., if x contains only one selected service and op_{div} is a delete service operator), the operator is not applied at all. For $S \geq 1$, the operator is applied in Step 12. The solution after the diversification step is saved in the variable x_{div}.

The intensification loop includes Steps 13– 16 and looks only at the timetabled services m that have been added or shifted by the diversification operator. The intensification operator to be applied is selected in Step 14 and the number S of stations to be modified in Step 15. The selection logic of S is similar to Step 11, this time the upper bound $\mathcal{S}(x, m, \text{op}_{\text{int}}, S_{\max})$ ensures it is indeed possible to add, delete or shift S stations in service m. After applying the respective operator in Step 16, the objective value of x is compared to the objective value of x_{div} in Step 17. If the intensification has decreased the objective value, the solution x_{div} becomes the current solution in Step 18.

In the non-sequential case, an operator op_r is selected randomly in Step 20 based on the probabilities π_{op}. We need to differentiate based on the *purpose* of the operator, which is determined in Step 21: If it is a diversification operator, we can directly select the value S in Step 22 with an identical logic to Step 11 and apply the operator in Step 23. Yet, if it is an intensification operator, we first determine the service to be adjusted in Step 25 and then select S in Step 26 with an identical logic to Step 15. Based on these selections, the operator is applied in Step 27.

Step 28 checks whether a final reoptimization is desired. If so, the loop comprising Steps 29–32 is traversed for every selected timetabled service m. Within the loop, the operator $7^{m,m}$ is applied in Step 30. Finally, if an improvement in the objective value is confirmed in Step 31, the solution x_{reopt} becomes the current and the best solution in Step 32.

5. Computational results

In this section, we present computational results based on the model (1)–(11b) and the solution algorithms introduced in Section 4. The computational setup, comprising the set of sample instances as well as the parameter settings, is presented in Section 5.1. In the following, we analyze setups for the LMNS algorithm that lead to a favorable tradeoff between good results and fast computation times in Section 5.2. We compare the results of the metaheuristics with specific setups where optimal solutions are known in Section 5.3. Subsequently, we discuss the benefits of the innovative model aspects in Section 5.4 and provide an example of a model extension in Section 5.5.

5.1. Computational setup

Our computational results are based on an extension of the 30 instances introduced in (Steiner and Irnich, 2018). The instances are summarized in Table 4 and an instance with the respective properties is included in our experiments if and only if it is marked by a +. As before, the

Algorithm 1: LMNS algorithm

Input: Probabilities $\{\pi_{op}\}$
Setup parameters *sequential_ approach* and *final_ reopt*
Acceptance criterion AccCr
Parameters $It_{\text{LMNS}}, \rho, S_{\max}, \epsilon_0$

$x := x^{accepted} := x^{best} :=$ InitialSolution()
for $iter := 1, \ldots, It_{\text{LMNS}}$ **do**
  if $obj(x) > obj(x^{best})$ **then**
    $x^{best} := x$
  if AccCr$(x, x^{best}, \epsilon_0, iter)$ **then**
    $x^{accepted} := x$
  else
    $x := x^{accepted}$
  if *sequential_ approach* $=$ *true* **then**
    Randomly choose diversification operator op_{div} according to weights $\{\pi_{\text{op}}\}$
    Randomly choose a value for S with $1 \leq S \leq \mathcal{S}(x, \text{op}_{\text{div}}, S_{\max})$
    $x_{\text{div}} := x := \text{op}_{\text{div}}^{S,\rho}(x)$
    for each added or shifted service m **do**
      Randomly choose intensification operator op_{int} according to weights $\{\pi_{\text{op}}\}$
      Randomly choose a value for S with $1 \leq S \leq \mathcal{S}(x, m, \text{op}_{\text{int}}, S_{\max})$
      $x := \text{op}_{\text{int}}^{S,m,\rho}(x)$
    if $obj(x_{\text{div}}) > obj(x)$ **then**
      $x := x_{\text{div}}$
  else
    Randomly choose operator op_r according to weights $\{\pi_{op}\}$
    if op_r is a diversification operator **then**
      Randomly choose a value for S with $1 \leq S \leq \mathcal{S}(x, \text{op}_r, S_{\max})$
      $x := \text{op}_r^{S,\rho}(x)$
    else
      Randomly choose service m out of selected services
      Randomly choose a feasible value for S with $1 \leq S \leq \mathcal{S}(x, m, \text{op}_r, S_{\max})$
      $x := \text{op}_r^{S,m,\rho}(x)$
if *final_ reopt* $=$ *true* **then**
  for each selected service m **do**
    $x_{\text{reopt}} := 7^{m,m}(x)$
    if $obj(x_{reopt}) > obj(x)$ **then**
      $x^{best} := x := x_{\text{reopt}}$

characteristics of the instances differ in three dimensions: First, the *number of cities* where the bus can stop with the options of 12, 15, and 18 cities. Second, the *corridor* in which the cities are located with four different corridors examined. Since corridor 4 is the smallest corridor, there are no instances with 15 and 18 cities in it. Third, the demand *scenarios* with a baseline scenario `Base`, a conservative scenario `Cons`, and an optimistic scenario `Opti`.

	12 cities			15 cities			18 cities		
Scenario	`Base`	`Cons`	`Opti`	`Base`	`Cons`	`Opti`	`Base`	`Cons`	`Opti`
Corridor 1	+	+	+	+	+	+	+	+	+
Corridor 2	+	+	+	+	+	+	+	+	+
Corridor 3	+	+	+	+	+	+	+	+	+
Corridor 4	+	+	+	−	−	−	−	−	−

Table 4: Instances for computational results

The underlying demand inputs are again based on the customized model developed in cooperation with our industry partner for the computational study in (Steiner and Irnich, 2018). The remaining parameters are chosen as follows in line with our previous approach: We split the day in ten departure time intervals T_k (nine intervals with two hour duration each and one interval from 12 a.m./midnight to 6 a.m.) and the potential start time of the m-th timetabled service is chosen to coincide with the beginning of the interval T_k, i.e., $c_m = a_{m-1}$. Further, we have 14 duration intervals D_l (one interval for travel times up to 60 minutes, six intervals in 30 minute steps up to four hours, six intervals in 60 minute steps up to ten hours and one interval for longer trips). Travel distances, travel times, ticket prices, and variable costs have been chosen identically to the study in (Steiner and Irnich, 2018). We exclude fixed costs because commercial agreements with transportation suppliers are usually based on a price per kilometer. The maximum number of timetabled services is $F = 3$, since demand model calibration was only possible in this range based on the real-world data of our industry partner. Finally, capacities are chosen as $C_m = 52$ and the auxiliary parameter as $u = 1$ minute.

All computational tests are performed on a standard PC with an Intel(R) Core(TM) i7-2600 running at 3.4 GHz with 16 GB of main memory using a single thread. Algorithms are coded in C++ using CPLEX 12.7 and compiled in release mode with MS Visual Studio 2015.

5.2. Technical aspects

In this section, we determine an operator selection strategy and parameter settings for the LMNS algorithm that allow for a good solution quality in acceptable computation time. As discussed in Section 4.2, the different operator selection strategies can be clustered by the *probabilities* of the operators, by the application of a *sequential approach* for diversification and intensification, as well as by the inclusion of a *final re-optimization.*

To simplify the ample possibilities for different operator probabilities, we start with clusters based on the three *degrees of randomness* of the operators. Specifically, for the diversification and intensification operators, we either include all (indicated by +) or none (indicated by −) of the operators of each degree of randomness, which yields seven strategies. These are presented in the top seven rows for each number of cities in Table 5, e.g., (+ − −) indicates the strategy of only including the random operators. Identical probabilities are assigned to every operator that is included. Recall the operator 4.3 *Optimized add best services* is not included in the LMNS, hence we replace it by the operator 4.2 in the strategy that only includes the optimized best operators to ensure there is a possibility to add a service. Due to promising results during the pre-tests, we analyze one more strategy denoted by ($+_{\text{div}}\ -\ +_{\text{int}}$), which includes only the random operators for the diversification and only the optimized best operators for the intensification.

Furthermore, the three matheuristics introduced in Section 4.2, which are based on the optimization model from (Steiner and Irnich, 2018), are included. For the matheuristics, the number of timetabled services to be selected needs to be fixed before running the algorithm. Therefore, we run them with $F = 1, 2, 3$ and use the value F with the best results per instance for the gap statistics. Option c requires intensification operators to be included as defined in Section 4.2. We choose to include only the optimized best intensification operators as they showed the best results during the pre-tests.

The LMNS is run with the following initial parameter settings based on the pre-tests: There are $It_{\text{LMNS}} = 5,000$ iterations and in each operator application only one station or timetabled service is adjusted, i.e., $S_{\max} = 1$. Further, solutions with up to $\epsilon_0 = 5\,\%$ decrease in objective value are accepted at the beginning of the algorithm and we use the exponent $\rho = 10$ for the randomization. For each instance, we run the LMNS ten times, each time with a different initial random seed. The matheuristics based on operator 7 show much less variation for different random seeds, we therefore decided to run these algorithms only three times.

We tried to solve the smaller instances with the model (1)–(11b), however we did not obtain any solutions even after several hours of computation time. For the instance with 12 cities, demand scenario `Base`, corridor 1, and $F = 2$, the gap after one hour (two hours) was $112.2\,\%$ ($110.9\,\%$) and the best solution was $9.0\,\%$ ($9.0\,\%$) worse than the best solution found with the LMNS algorithms. Therefore, the following experiments are based only on the LMNS algorithms.

Table 5 presents the results with respect to the gaps and computation times. We separate the results by the number of cities to understand the impact of increasing instance size. *Gap Avg.* represents the average gap to the best solution per instance, that we obtained in this experiment. I.e., the average is taken over the 12 instances with 12 cities (9 instances in case of 15 and 18 cities) as well as the ten different LMNS runs (three in case of the matheuristics). Meanwhile, *Gap Best* is calculated based on the minimum gap per instance and subsequently taking the average over the 12 (or 9) instances. The average computation time in seconds is presented in *Time Avg*, for the matheuristics this is based on the sum of the setups with $F = 1, 2, 3$.

We observe that the non-sequential setups perform consistently better than the sequential setups. Therefore, we omitted the sequential tests for the 18 city instances. The poor results of the sequential setup could be an indicator that the diversification operators should not be applied in every iteration. Hence, we test non-sequential setups with lower weights for the intensification operators in the following experiments.

Regarding the final reoptimization, there are consistent improvements, however they are smaller in the non-sequential setup. As a consequence, we recommend only using this setup in case sufficient time is available and minor improvements of the objective value are critical. Beyond that, approaches could be explored which apply the reoptimization of the best solution multiple times, e.g., after every $1,000$ iterations.

The strategies based on the matheuristics perform consistently inferior to the other approaches, further examinations show the algorithms frequently get stuck in local fixed points. Based on this observation, we did not test the matheuristics for the 18 city instances.

The general results regarding the operator selection are similar for the different instance sizes.

	Oper. selection			*Gap Avg. (%)*				*Gap Best (%)*				*Time Avg. (s)*			
	Included clusters			non-sequential		sequential		non-sequential		sequential		non-sequential		sequential	
number of cities	Ran-dom	Heu-ristic	Opti-mized	no final reopt	final reopt	no final reopt	final reopt	no final reopt	final reopt	no final reopt	final reopt	no final reopt	final reopt	no final reopt	final reopt
12	+	−	−	2.2	2.2	9.2	6.2	1.0	0.9	3.8	2.1	2.0	4.8	3.1	6.0
	−	+	−	4.7	4.7	9.9	7.1	2.1	2.1	3.9	2.1	2.2	4.8	3.1	5.9
	−	−	+	4.2	4.2	7.5	5.2	0.2	0.2	2.8	2.2	13.3	15.9	23.3	26.2
	+	+	−	2.7	2.6	8.7	6.0	1.0	0.9	3.3	2.3	2.2	4.8	3.2	6.0
	+	−	+	2.2	2.0	7.1	5.1	0.9	0.8	2.7	1.9	8.1	10.9	13.1	15.9
	−	+	+	3.5	3.5	7.9	5.8	1.0	0.9	2.6	2.0	8.0	10.6	12.3	15.0
	+	+	+	1.9	1.9	6.9	4.7	0.9	0.8	1.9	1.2	6.2	9.0	9.8	12.6
	$+_{\text{div}}$	−	$+_{\text{int}}$	1.8	1.8	6.9	4.6	1.0	0.9	3.2	1.9	9.3	12.0	12.0	15.0
	Math. option a			8.9	−	−	−	8.9	−	−	−	663.7	−	−	−
	Math. option b			7.9	−	−	−	7.2	−	−	−	596.0	−	−	−
	Math. option c			−	−	6.9	−	−	−	5.6	−	−	−	656.9	−
15	+	−	−	5.0	4.7	13.0	10.8	1.7	1.5	5.7	4.7	3.3	30.7	4.7	38.4
	−	+	−	8.7	8.1	13.7	11.1	5.1	4.1	7.5	6.2	3.9	32.8	4.7	35.8
	−	−	+	3.8	3.7	10.0	8.4	0.7	0.7	4.1	3.4	30.6	57.7	45.3	77.2
	+	+	−	6.0	5.5	11.7	9.6	3.4	3.1	5.0	3.4	3.7	31.7	4.7	36.2
	+	−	+	3.3	3.3	9.6	7.8	1.2	1.2	3.5	2.9	16.4	42.1	26.3	58.9
	−	+	+	4.5	4.4	8.6	7.6	1.2	1.2	3.5	3.2	17.8	43.7	23.6	55.4
	+	+	+	3.8	3.7	9.5	8.2	0.8	0.8	3.2	2.6	12.5	38.6	17.4	49.3
	$+_{\text{div}}$	−	$+_{\text{int}}$	3.2	3.2	9.9	8.6	0.6	0.6	4.5	3.7	22.9	51.0	23.3	57.9
	Math. option a			12.5	−	−	−	11.6	−	−	−	5,681.7	−	−	−
	Math. option b			11.6	−	−	−	9.9	−	−	−	5,204.8	−	−	−
	Math. option c			−	−	10.1	−	−	−	8.4	−	−	−	5,612.1	−
18	+	−	−	6.9	−	18.2	−	2.8	−	9.9	−	4.5	−	6.0	−
	−	+	−	9.2	−	16.6	−	4.9	−	8.6	−	5.4	−	6.6	−
	−	−	+	4.4	−	11.7	−	0.0	−	6.1	−	53.0	−	68.9	−
	+	+	−	6.4	−	14.6	−	2.3	−	5.8	−	5.1	−	6.6	−
	+	−	+	3.9	−	12.9	−	1.9	−	5.4	−	28.8	−	39.7	−
	−	+	+	5.7	−	10.3	−	1.3	−	4.0	−	28.4	−	36.6	−
	+	+	+	4.2	−	10.8	−	1.1	−	4.4	−	21.6	−	27.7	−
	$+_{\text{div}}$	−	$+_{\text{int}}$	2.6	−	12.2	−	0.5	−	3.7	−	43.1	−	40.5	−

Table 5: Computation results for different operator selection strategies for the LMNS algorithm

The setup $(- - +)$ performs best with respect to *Gap Best* and $(+_{\text{div}} - +_{\text{int}})$ with respect to *Gap Avg.* While the heuristic operators perform nearly as fast as the random operators, the gaps are high compared to the other strategies. The underlying reason could be the lack of foresight regarding the trip frequency and cannibalization effects on the passenger numbers after the multi-commodity network-flow model is solved.

In summary, the most promising setup, which serves as baseline setup for further parameter studies, is based on the non-sequential setup, no final reoptimization, and selecting only the heuristic best operators. To reflect the strong average results of the setup $(+_{\text{div}} - +_{\text{int}})$, we also test variations of this setup with lower weights on the intensification operators. The strategy with only 5 % of the total probabilities for the diversification operators is denoted by $(+^{5}_{\text{div}} - +^{95}_{\text{int}})$ and identical probabilities are chosen among the diversification and intensification operators, respectively. Table 6 presents the results when varying the parameters $It_{\text{LMNS}}, \rho, S_{\max}, \epsilon_0$, and when including adjusted versions of the setup $(+_{\text{div}} - +_{\text{int}})$. All studies are conducted based on a "ceteris paribus" approach, i.e., only one parameter is adjusted at a time while the others take their initial values used for the study above.

Parameter setup	*Gap Avg. (%)*			*Gap Best (%)*			*Time Avg. (s)*		
cities	12	15	18	12	15	18	12	15	18
Baseline $-/-/+$	4.2	3.8	4.4	0.2	0.7	0.0	13.3	30.6	52.8
$It_{\text{LMNS}} = 1,000$	5.3	5.0	5.4	0.3	1.7	1.5	2.9	6.6	11.1
$It_{\text{LMNS}} = 50,000$	3.9	3.3	3.7	0.2	0.7	0.0	129.8	300.0	523.1
$\rho = 5$	3.6	3.1	3.7	0.8	0.9	0.4	13.2	30.1	53.7
$\rho = 20$	4.0	4.1	4.3	0.9	1.1	0.2	13.0	29.5	51.2
$S_{\max} = 3$	3.7	2.7	3.3	0.9	1.1	0.3	19.2	48.9	89.5
$S_{\max} = 5$	3.7	2.7	3.3	0.9	1.1	0.3	19.2	48.9	89.4
$\epsilon_0 = 1\,\%$	6.4	7.9	7.6	1.6	2.8	2.4	13.0	30.6	51.3
$\epsilon_0 = 15\,\%$	2.2	3.7	5.2	1.5	1.3	0.7	12.3	28.0	49.3
$(+^{5}_{\text{div}} - +^{95}_{\text{int}})$	3.1	5.0	5.9	0.0	0.2	1.0	14.9	33.0	59.6
$(+^{25}_{\text{div}} - +^{75}_{\text{int}})$	2.3	3.4	2.6	0.0	1.0	0.4	12.7	29.2	56.2

Table 6: Computation results for parameter studies for the LMNS algorithm

We observe that the only setup that consistently decreases the gaps is $It_{\text{LMNS}} = 50,000$. A general trend that can be observed is that increasing randomness (smaller ρ, bigger $S_{\max}$, higher ϵ_0, and including random operators) improves the results for *Gap Avg. (%)*, while *Gap Best (%)* deteriorates. We choose $(+^{25}_{\text{div}} - +^{75}_{\text{int}})$ with $It_{\text{LMNS}} = 50,000$ for the following experiments as it consistently improves *Gap Avg. (%)* while only partly deteriorating the *Gap Best (%)*. Finally, Table 7 shows the results when removing a single operator from the setup $(+^{25}_{\text{div}} - +^{75}_{\text{int}})$. It can be observed that all options for removing an operator do not yield consistently better results than the setup with all six operators applied. Hence, every operator is useful for the LMNS algorithm and is included in the next sections.

5.3. Solution quality of the LMNS algorithm

A priori, there is no knowledge about the potential delta between the best known solutions and the optimal solutions. Therefore, to understand the quality of the solutions found with the LMNS

Parameter setups	Gap Avg. (%)			Gap Best (%)			Time Avg. (s)		
cities	12	15	18	12	15	18	12	15	18
$(+^{25}_{\text{div}} - +^{75}_{\text{int}})$	2.3	3.3	2.6	0.0	1.0	0.4	12.7	29.2	56.2
Removed op. 1.3	19.1	19.2	19.8	8.4	11.0	13.1	10.1	18.8	30.0
Removed op. 2.3	10.6	19.9	23.2	3.9	14.0	14.5	12.2	36.7	80.7
Removed op. 3.3	3.1	4.0	4.5	0.0	1.4	1.1	7.0	13.1	20.7
Removed op. 4.1	8.8	30.9	36.8	0.2	8.2	14.7	10.4	18.9	32.8
Removed op. 5.1	7.3	2.0	1.8	3.1	0.1	0.2	17.1	34.6	61.5
Removed op. 6.1	5.7	9.2	6.7	0.1	1.4	0.7	13.0	31.2	55.6

Table 7: Computation results when removing single operators from LMNS algorithm

algorithm, it is not sufficient to look at the gaps to the best known solutions. However, we are able to determine a solution of proven optimality in the special case, where the demand used in the model does not behave dynamically with respect to trip frequencies and cannibalization. In this case, the optimal solution can be determined based on the existing optimization model for single timetabled services because the problem decomposes into sub-problems, one for each potential timetabled service m. To determine the optimal solution with F selected timetabled services, we can simply pick the F best services as the objective values of the services are independent of each other. In the following experiments, the demand parameters are based on constant trip frequencies and no cannibalization is considered. Table 8 shows the average and best gaps per number of cities when running the LMNS with the settings determined above.

Number of services	Gap Avg. (%)			Gap Best (%)		
cities	12	15	18	12	15	18
$F = 1$	0.02	0.07	0.00	0.00	0.00	0.00
$F = 2$	0.01	0.00	0.05	0.00	0.00	0.00
$F = 3$	0.00	0.00	0.10	0.00	0.00	0.00

Table 8: Solution quality of LMNS results with respect to known optimal solutions:
No dynamic demand with respect to trip frequency and cannibalization

When including the trip frequency and cannibalization effects, we can solve another set of instances to optimality: For 12 cities and a maximum number of services $F = 2$, the model can be solved with an enumeration in several hours of computation time. Table 9 shows the average and best gaps per instance when the best results of the LMNS algorithm are compared against the optimal solutions. Indeed, the optimal solution could be found in 11 out of 12 cases.

In summary, the LMNS algorithm found the optimal solution for 101 out of the 102 instances with known optimal solution. Based on these promising results, it is reasonable to assume that the optimal solutions obtained with the LMNS algorithm are of very good quality also in cases with an unknown optimal solution.

Scenario	Base				Cons				Opti			
Corridor	1	2	3	4	1	2	3	4	1	2	3	4
Gap Avg. (%)	2.91	1.43	8.18	0.64	0.41	0.42	1.91	0.42	0.00	0.00	1.80	0.00
Gap Best (%)	0.00	0.00	0.00	0.00	0.00	0.00	1.68	0.00	0.00	0.00	0.00	0.00

Table 9: Solution quality of LMNS results with respect to known optimal solutions:
12 city instances with $F = 2$

5.4. Modeling scope

The benefits of the schedule-based approach and the dynamic demand with respect to travel duration and departure time are discussed in Section 5.2.3 of (Steiner and Irnich, 2018). Hence, we focus on the additional aspects trip frequency and cannibalization in the following experiments. As a first step, we show that solutions differ indeed significantly when the impact of trip frequency and cannibalization on the demand are considered. To ensure the results we present are not skewed due to non-optimal solutions found by the LMNS, we base them on the instances with 12 cities, where the optimal solution can be determined based on an enumeration algorithm. For the instance with 12 cities, demand scenario `Base`, corridor 1, and $F = 2$, Table 10 shows the optimal solutions once considering the impact of trip frequency and cannibalization on the demand, and once ignoring it.

		Station (included +/not included −)											
Dynamic demand setup	Selected service	1	2	3	4	5	6	7	8	9	10	11	12
Impact of trip frequency and cannibalization considered	$m = 2$	+	+	−	+	+	−	+	−	+	+	+	+
	$m = 4$	+	+	+	−	−	−	+	+	−	+	−	+
Impact of trip frequency and cannibalization not considered	$m = 2$	+	+	+	+	−	+	+	−	+	+	+	+
	$m = 3$	+	+	+	+	−	−	+	−	+	+	+	+

Table 10: Impact on the optimal solutions of including trip frequency and cannibalization dynamics

We observe that the two optimal solutions differ significantly: Already the selected timetabled services are different. On the one hand, the setup ignoring the cannibalization chooses "neighboring" services with very similar selected stations (only for station 6 there is a difference). On the other hand, the solution when considering dynamic demand effects with respect to trip frequency and cannibalization leaves more time between the two departures. Furthermore, it shows more differences between the two services to realize a higher coverage of stations with more attractive travel times and less cannibalization.

As a second step, we analyze sensitivity with respect to the degree of cannibalization in more detail. For the instance with 12 cities, demand scenario `Opti`, corridor 1, and $F = 2$, we look at different intensities of cannibalization, parametrized by $\sigma \in \{0, 1, \ldots, 10\}$. For the degree of cannibalization g_1, the respective demand parameter for the case of no cannibalization (g_0) is multiplied by $1-\frac{\sigma}{40}$, for g_2 with $1-\frac{\sigma}{20}$, respectively. This means, for $\sigma = 0$ there is no cannibalization effect at all, while in the other extreme case of $\sigma = 10$ only 75 % of the original demand remain for a degree of cannibalization g_1, and 50 % for g_2. To isolate the effect of the varying cannibalization

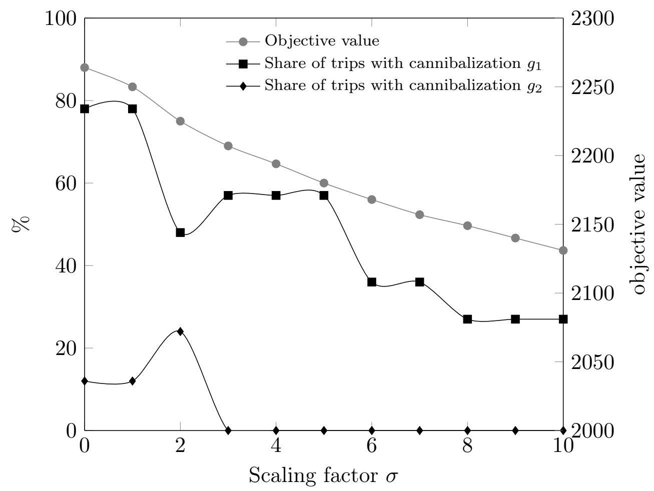

Figure 1: Sensitivity of best known solutions towards the intensity of cannibalization

level, the solution needs to select exclusively the services $m = 3$ and $m = 4$ and no dynamic demand with respect to the trip frequency is assumed. Table 11 shows the optimal solutions for the extreme cases $\sigma = 0$ and $\sigma = 10$. Complementary, Figure 1 shows further key characteristics of the solutions for varying values of σ.

		Station (included +/not included −)											
Scaling parameter	Selected service	1	2	3	4	5	6	7	8	9	10	11	12
$\sigma = 0$	$m = 3$	+	+	+	−	+	+	+	+	+	+	+	+
	$m = 4$	+	+	+	−	+	−	+	+	+	+	+	+
$\sigma = 10$	$m = 3$	+	+	−	+	−	−	+	−	+	+	+	+
	$m = 4$	+	+	+	−	+	+	+	+	+	+	+	+

Table 11: Sample best solutions for different intensities of cannibalization

The observations are in line with the expected results: The selected stations of the two services are very similar for $\sigma = 0$ and differ substantially for $\sigma = 10$. Further, the share of offered trips with a degree of cannibalization g_1 or g_2 as well as the achievable objective value decline with increasing values of σ.

Overall, we can conclude that considering the interdependencies of selected timetabled services has a significant impact on optimal solutions. It is therefore advisable for bus operators to consider these aspects in their demand modeling and applying models capable of incorporating these aspects.

5.5. *Model extensions*

In the final set of experiments, we analyze the effect of requesting services to be equal or similar as motivated in Section 3.2. For the instance with 15 cities, demand scenario `Cons`, corridor 1, and $F = 3$, we run the LMNS algorithm three times: once requiring all selected services to include identical selected stations (*identical services*), once again with the initial requirement of identical services but allowing for one additional station per service (*additional station per service*), and once with no additional requirements on the structure of the different services (*no requirements*). For the setup with *identical services*, the LMNS operators need to be slightly adjusted to ensure the services are still identical after an operator application. Specifically, the adjusted operators 1.3, 2.3, and 3.3 simultaneously add, delete or shift the best station (randomized as before) in all services. Further, operator 4.1 only selects the timetabled service to be added, the stations are then selected exactly as for the other selected services. The operators 5.1 and 6.1 need no adjustment, since they already preserve the property of identical services.

When an *additional station per service* is allowed, we proceed analogously to the case with identical services and include a final iteration applying operator 1.3 *optimized add best stations* for $S = 1$ to each selected service. The solution is accepted in case of an improved objective value.

Table 12 presents the three different best solutions found with the LMNS settings determined above. It can be observed that allowing an additional station per service can already improve the objective significantly. Yet, the solution with no requirements on the structure of the different lines is by far the most attractive one. Based on this type of analysis, the costs of additional requirements on the structure of the timetabled services can be determined.

		Station (included +/not included −)															
Structure requirements	Selected service	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	objective value
identical services	$m = 2$	+	−	−	+	+	−	+	+	−	−	−	+	+	+	+	
	$m = 4$	+	−	−	+	+	−	+	+	−	−	−	+	+	+	+	1,763
	$m = 5$	+	−	−	+	+	−	+	+	−	−	−	+	+	+	+	
additional station per service	$m = 2$	+	−	−	+	+	−	+	+	+	−	−	+	+	+	+	
	$m = 4$	+	−	−	+	+	−	+	+	−	−	−	+	+	+	+	1,860
	$m = 5$	+	−	−	+	+	−	+	+	−	−	−	+	+	+	+	
no requirements	$m = 2$	+	+	+	+	−	−	+	−	−	−	−	+	+	−	+	
	$m = 3$	+	−	+	−	+	−	+	+	−	+	−	−	+	+	+	2,633
	$m = 4$	+	+	+	−	−	−	+	−	+	−	−	−	−	−	+	

Table 12: Sample best known solutions for different requirements on the structure of services

6. Conclusion and outlook

We have presented a scheduled-based mixed-integer linear model formulation comprising multiple aspects of dynamic demand. This model can be applied by an operator of inter-city buses for the concurrent planning of multiple timetabled services. Simultaneous decisions are made on the characteristics of the network design, by selecting stations, and also on scheduling aspects, by selecting departure times. Since computation times that rely on standard approaches are too long to solve real-world instances, we have introduced different variations of a large multiple neighborhood search (LMNS) metaheuristic algorithm. In an extensive computational study we obtained solutions in attractive computation times and observed that the gaps to optimal solutions are small for the cases with known optimal solutions. Furthermore, we studied the modeling scope and discussed how considering the interdependencies between different timetabled services significantly impacts the optimal solutions.

Future research could focus on extending the scope from a single travel corridor to considering the entire network, including passenger transfers. This would require any future model to include even more interdependencies between services, since the demand for timetabled services in one corridor can depend on the offering of services in another corridor. The application of optimization models would therefore be connected even more closely to the development of demand models capable of predicting demand effects with high accuracy.

Another important research direction is the inclusion of further operational aspects when planning the timetabled services. Driver costs are an important aspect, since they represent a significant share of the operating costs and are typically associated with complex regulations, including in relation to working and driving times. Specifically, the variable costs could not be determined a priori for a connection between two stations. They would depend on both, the total structure of the timetabled service itself due to necessary breaks, and also on the other selected timetabled services. The latter aspect stems from the fact that only some selected pairs of services can be driven consecutively by the same driver.

We believe that metaheuristics will be the most suitable approach to tackle these complex problems and that the approaches and studies presented in this paper provide a solid foundation for the future work in this field.

References

Abdelghany, A., Abdelghany, K., and Azadian, F. (2017). Airline flight schedule planning under competition. *Computers & Operations Research*, **87**, 20–39.

Ball, M. O. (2011). Heuristics based on mathematical programming. *Surveys in Operations Research and Management Science*, **16**(1), 21–38.

Barrena, E., Laporte, G., Coelho, L., and Ortiz, D. (2013). *A Fast and Efficient Adaptive Large Neighborhood Search Heuristic for the Passenger Train Timetabling Problem with Dynamic Demand*. CIRRELT (Collection). CIRRELT, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport.

Canca, D., De-Los-Santos, A., Laporte, G., and Mesa, J. A. (2017). An adaptive neighborhood search metaheuristic for the integrated railway rapid transit network design and line planning problem. *Computers & Operations Research*, **78**, 1–14.

Cascetta, E. and Coppola, P. (2016). Assessment of schedule-based and frequency-based assignment models for strategic and operational planning of high-speed rail services. *Transportation Research Part A: Policy and Practice*, **84**, 93–108.

Desaulniers, G. and Hickman, M. D. (2007). Public transit. In C. Barnhart and G. Laporte, editors, *Transportation*, volume 14 of *Handbooks in operations research and management science*, chapter 2, pages 69–127. Elsevier BV.

Fockenbrock, D. and Heide, D. (2017). Next stop monopoly. *Handelsblatt*. `https://global.handelsblatt.com/companies/next-stop-monopoly-746100`.

Hassannayebi, E. and Zegordi, S. H. (2017). Variable and adaptive neighbourhood search algorithms for rail rapid transit timetabling problem. *Computers & Operations Research*, **78**, 439–453.

Hintsch, T. and Irnich, S. (2018). Large multiple neighborhood search for the clustered vehicle-routing problem. *European Journal of Operational Research*, **270**(1), 118–131.

Ibarra-Rojas, O., Delgado, F., Giesen, R., and Muñoz, J. (2015). Planning, operation, and control of bus transport systems: A literature review. *Transportation Research Part B: Methodological*, **77**, 38–75.

Masson, R., Lehuédé, F., and Péton, O. (2013). An adaptive large neighborhood search for the pickup and delivery problem with transfers. *Transportation Science*, **47**(3), 344–355.

Ortúzar, J. d. D. and Willumsen, L. G. (2011). *Modelling Transport*. John Wiley & Sons, Ltd.

Pisinger, D. and Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research*, **34**(8), 2403–2435.

Pisinger, D. and Ropke, S. (2010). Large neighborhood search. In M. Gendreau and J.-Y. Potvin, editors, *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research and Management Science*, pages 399–419. Springer US.

Ropke, S. and Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, **40**(4), 455–472.

Santini, A., Ropke, S., and Hvattum, L. M. (2018). A comparison of acceptance criteria for the adaptive large neighbourhood search metaheuristic. *Journal of Heuristics*, **24**(5), 783–815.

Schöbel, A. (2011). Line planning in public transportation: models and methods. *OR Spectrum*, **34**(3), 491–510.

Schöbel, A. (2017). An eigenmodel for iterative line planning, timetabling and vehicle scheduling in public transportation. *Transportation Research Part C: Emerging Technologies*, **74**, 348–365.

Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In M. Maher and J.-F. Puget, editors, *Principles and Practice of Constraint Programming — CP98*, volume 1520 of *Lecture Notes in Computer Science*, pages 417–431. Springer Berlin Heidelberg.

Steiner, K. and Irnich, S. (2018). Schedule-based integrated intercity bus line planning via branch-and-cut. *Transportation Science*, **52**(4), 882–897.