Submitted version from Dec 2016. Published version DOI: 10.1002/net.21824
Cite this as:
D. Lückerath, O. Ullrich, N. Rishe, and E. Speckenmeyer, A disjunctive program formulation to generate public transit timetables adhering to prioritized planning requirements, Networks (2018).

A disjunctive program formulation to generate regular public transit timetables adhering to prioritized planning requirements

Daniel Lückerath[1*], Oliver Ullrich[1], Naphtali Rishe[2], Ewald Speckenmeyer[3]

[1]Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS, Schloss Birlinghoven, 53757 Sankt Augustin, Germany
*daniel.lueckerath@iais.fraunhofer.de

[2]NSF Industry-University Cooperative Research Center, Florida International University, ECS 243C, 11200 SW 8th St, Miami FL-33199

[3]Department of Computer Science, University of Cologne, Weyertal 121, 50931 Cologne, Germany;

Keywords: Public transit, optimization, timetabling, scheduling, regularity, disjunctive programming, mixed integer programming

Abstract

Timetable regularity is an important measure for service quality in high frequency public transit systems, assuring an evenly distributed passenger load as well as improving product attractiveness and appreciation. However, to be feasible during daily operation a timetable may also have to adhere to other planning requirements, e.g. limitations to permitted departure times to better coordinate with other service providers like schools or long-distance transportation networks.

In this paper a disjunctive program formulation is proposed to generate regular timetables adhering to planning requirements. The modeled requirements not only allow for the consideration of feasibility constraints from daily operations, but also for the consideration of simultaneous departures for transfer connections, an objective traditionally opposed to regularity. To show its applicability the approach is applied to two models of artificial transit networks as well as to a model of the bus network of Cologne, Germany.

The results show that the proposed formulation can be used to generate timetables for network instances of realistic size in acceptable time using CPLEX. For networks consisting of multiple connected components it is shown that a divide and conquer approach can significantly reduce run times.

1. Introduction

Timetable generation is a complex task and a well-known optimization problem (see [6]). It can be subject to different objectives, ranging from achieving good transfer connections, to equidistant headways, to robustness in general. However, to be feasible during daily operation a timetable may also have to adhere to other planning requirements, e.g. specific departure sequences to accommodate frequent transfer connections. A significant portion of published optimization approaches for timetable generation seems to ignore such planning requirements and the resulting timetables subsequently may need to be further modified by hand to be feasible during daily operations.

In this paper a disjunctive program formulation is proposed, to generate regular timetables adhering to planning requirements. The modeled requirements not only allow for the consideration of feasibility constraints from daily operations, but also for the consideration of simultaneous departures for transfer connections, an objective traditionally opposed to regularity. To show its applicability the approach

is applied to two models of artificial transit networks as well as to a model of the bus network of Cologne, Germany.

The remainder of this paper is organized as follows: section 2 gives a brief overview of the background of timetable generation as well as related research. Section 3 deals with the problem formulation, a first basic description of a mixed integer program to generate regular timetables, and an extension of this model to incorporate planning requirements. In section 4 the applicability of the model is shown, before section 5 concludes the paper with a summary of the results and some proposals for further research directions.

2. Background

2.1. Timetable generation

Timetable generation is the process of combining the results of the network and line planning phase with the defined service frequencies to transform them into a feasible schedule, i.e. a set of trips with departure times (see [6]). What exactly makes a timetable feasible depends, amongst other things, on the transit system under consideration, the potential passenger group to be targeted, and other operational, political, and societal conditions. As a result, there exist different equally suitable objectives for timetable generation.

A frequently used objective is *service regularity* (see e.g. [1], [9], [11], [12], [17], and [20]), a measure for the equability of headways. It can be used for static evaluation of a timetable during the planning phase as well as for dynamic assessment of operational performance. As shown by van Ort and van Nes in [20], a timetable with high regularity during daily operation results in a more even distribution of passengers across vehicles, lowering spikes in passenger load and making the transit system more attractive. Ullrich et al. showed in [19] that in systems where vehicles of different lines share common resources with limited capacity (i.e. most tram systems), regularity can also be used as a measure for the capability of a timetable to contain consequences of small delays, subsequently heightening the overall punctuality of the system.

Another frequently used objective for timetable generation is *synchronization* (see e.g. [4], [8], [11], [12], [14], and [22]), which is a measure for the number of (almost) simultaneous vehicle departures/arrivals and as such is opposed to regularity. Timetables with a high synchronization help to assure attractive transfer connections for passengers by minimizing waiting time, which is especially important in transit systems with low service frequencies where passengers otherwise may need to wait for a prolonged time.

A third frequently used objective is *robustness* (see e.g. [2], [3], [10], [13], [15], and [23]), which is a universal measure for the capability of a system to withstand dynamically occurring disturbances without changes to its initial stable structure. Because of the generality of this objective, different interpretations can be employed when generating timetables. It is therefore necessary to specify precisely what type of disturbance is to be addressed when optimizing for robustness. For example, and as already explained, a timetable with high service regularity can raise robustness against small disturbances in transit systems where vehicles of different lines share common resources with limited capacity. In contrast, a timetable with buffer times between the planned arrival and departure times of vehicles can raise the robustness of transfer connections by allowing punctual or early vehicles to wait on feeder vehicles.

In addition to the objectives applied during their generation, timetables can be classified by their structural characteristics. These characteristics comprise, amongst others, the periodicity of the timetable (i.e. periodic or aperiodic) as well as the homogeneity of the basic intervals (i.e. one common basic interval for all lines or different basic intervals for different lines).

2.2. Related research

A multitude of different optimization approaches for timetable generation exist, which can be roughly distinguished by the level of abstraction of the optimization model, the applied solution method, the utilized (combination) of objectives, and the structural characteristics of the generated timetable (for a general overview see e.g. [6]).

A fundamental optimization approach was introduced by Ceder et al. in [4] and later extended by Eranki in [8]. It enables the generation of aperiodic timetables with maximum synchronization at dedicated transfer stops while simultaneously adhering to given minimum and maximum headways for vehicles of the same line. The model is based on a graph representation of the transit network with reduced sets of nodes and edges, i.e. it contains only nodes and edges associated with previously identified transfer stops. Ceder et al. then develop a mixed integer linear program (MILP) to maximize the number of simultaneous arrivals at the nodes of the graph. To solve the MILP the authors develop a constructive algorithm, which first identifies the most promising node at any given moment and then tries to schedule as many simultaneous arrivals there as possible. If no more (synchronous) arrivals can be scheduled, the next promising node is identified, and the process repeats. Ceder et al. show the correctness and applicability of their approach by applying it to models of artificial transit networks as well as to a model based on a part of the bus network of Israel.

More recent approaches for the generation of synchronous timetables are described by Saharidis et al. in [14] and Wu et al. in [22]. Saharidis et al. propose a MILP with several extensions to generate bus timetables, which minimize cost associated with passenger waiting time at transfer stops and also take phases with high passenger demand into account. They show the applicability of their model by using CPLEX to generate timetables for the bus network of the Greek island of Crete and comparing the simulated waiting times with the observed waiting times.

Wu et al. also develop a model for the minimization of passenger waiting times at transfer stops. But other than Ceder et al. and Saharidis et al. they create periodic timetables and assume stochastic driving and waiting times. Additionally, Wu et al. differentiate between waiting time of passengers newly boarding a vehicle at a transfer stop, waiting time of passenger transferring from one vehicle to another at a transfer stop and waiting time for passengers just passing through a transfer stop without changing vehicles. To mitigate the randomness of travel times they allocate buffer times to vehicle departures. To solve their model Wu et al. develop a Genetic Algorithm (see [7]) whose probabilistic search strategy they replace with a simple local search strategy to assure a quick and reliant convergence. They demonstrate the performance and applicability of their approach by generating timetables for randomly generated network instances and comparing the waiting times resulting from timetables generated using their approach with waiting times of timetables generated under the assumption of deterministic driving times. The results show that the usage of stochastic driving times during optimization results in timetables which are better suited to conditions found during daily operations: They show a reduction in average waiting time for newly boarding passengers as well as in average waiting time for passengers transferring between vehicles, while raising the average waiting time for passengers just passing the transfer stop.

Optimization models to generate regular timetables are for example proposed by Genç in [9], Bampas et al. in [1] and Ullrich et al. in [17] and [19]. Genç develops an integer linear programming formulation to generate periodic timetables with variable basic intervals, maximizing the minimum headway between consecutive departures at stops shared by multiple lines. This results in timetables which distribute all planned departures as equidistantly in their common basic interval as possible, thus raising the robustness of the system against small disturbances. Genç solves his model using a self-developed Branch-and-Bound solver (see [5]) and applies it to the generation of timetables for the tram network of Cologne, Germany. To assess the performance of his solver he compares it with CPLEX. The results show that the Branch-and-Bound solver quickly generates good solutions, but with increasing

run time CPLEX is able to find better solutions. Furthermore, Genç shows that the optimization problem of generating timetables with maximum service regularity can be traced back to the edge coloring problem, and thus is an NP hard optimization problem.

Bampas et al. develop exact and heuristic methods to generate periodic regular timetables for special cases of the optimization problem posed by Genç, i.e. they generate timetables with a common basic interval for special network structures, like chains, spiders, rings, and trees. As Genç, Bampas et al. show that the underlying optimization problem can be traced back to the edge coloring problem of graph theory and prove that the generation of maximum regularity timetables with common basic interval is NP hard for ring and tree networks. The authors develop exact algorithms for chain, star, and spider networks as well as heuristic algorithms to approximate the optimal solutions for ring and tree networks.

Ullrich et al. also solve a special case of the optimization problem formulated by Genç, addressing the generation of regular, periodic timetables with a common basic interval. They extend the model with the possibility to incorporate transit planning requirements (e.g. guaranteed transfer connections, amplifier lines serving only the city centers, coordination with intercity rail traffic) during optimization to assure the applicability of the resulting timetables during daily operation. To solve their model the authors develop a Branch-and-Bound solver, which is preceded by a Genetic Algorithm supplying initial upper bounds to kick start the solver. Ullrich et al. apply their approach to several models of the tram networks of Cologne, Germany and Montpellier, France and show that the resulting timetables raise the robustness against small disturbances, while simultaneously respecting formulated planning requirements.

A model combining regularity and synchronization is proposed by Ibarra-Rojas and Rios-Solis in [11] and extended by Ibarra-Rojas et al. in [12]. In [11] they develop an Integer Programming formulation to generate timetables, which maximize the number of synchronous departures to optimize passenger transfers, while also targeting regular departure times to avoid vehicle bunching. To solve the problem, they develop a multi-start iterated local search algorithm and apply it to different network models based on the bus network of Monterrey, Mexico. They compare timetables generated by their approach with timetables obtained using CPLEX and find that their algorithm generates good solutions very quickly, while CPLEX does not solve to optimality in acceptable time.

In [12] the authors extend their model with the possibility to consider multiperiod synchronization, i.e. synchronization between vehicles departing in different planning periods of the operational day. To solve this problem Ibarra-Rojas et al. define several different neighborhood functions to search the solution space as well as different heuristic approaches using these neighborhoods to approximate solutions for their problem. They apply the resulting approaches to models of different artificial bus networks as well as to a model based on part of the bus network of Monterrey, Mexico and again compare the obtained solution with solutions computed using CPLEX. Their results show that their heuristics can compute good solutions very quickly and that they achieve a higher number of synchronous departures than approaches that merge separately optimized planning periods.

An approach for the generation of robust timetables is for example described by Bešinović et al. in [2] and Goverde et al. in [10]. Bešinović et al. describe a hierarchical two-level framework for the generation of robust railway timetables, which combines a macroscopic and a microscopic model in an iterative process. The macroscopic model generates timetables with minimal travel times and maximal robustness by solving an integer linear program for a reduced railway network (i.e. only locations that allow interactions between trains are considered) using a randomized multi-start greedy heuristic and evaluating them in regards to their ability to absorb delays via different scenarios with randomly generated delays. The resulting timetables are fed to the microscopic model, which represents the network in more detail, i.e. it considers all locations where section attributes (e.g. speed limit, gradient, or curvature radius) change. The model is used to calculate running, blocking, and minimum headway

times and to evaluate the timetables for potential train conflicts and infrastructure occupation violations. Based on these evaluations the constraints of the macroscopic model may be updated and a new iteration may be started. This process is repeated until a robust, stable, and conflict-free timetable is found.

The model is extended with a third level by Goverde et al.: a mesoscopic model for fine-tuning train trajectories on corridors between the macroscopic nodes of the network, in order to optimize energy-efficiency. After the iterations between the macroscopic and microscopic models are finished, the timetable is given to the mesoscopic model, which uses a dynamic programming approach to optimize the departure and arrival times at intermediate stops on corridors in order to lower the energy consumption of trains while travelling on connecting sections. Both Bešinović et al. and Goverde et al. apply their framework to a model of the Dutch railway network between Utrecht, Eindhoven, Tilburg, and Nijmegen. Their results show that their approach generates conflict-free timetables, which are more robust and energy efficient than the currently employed timetable.

3. Generating regular timetables adhering to planning requirements

In the following the MILP first described by Genç in [9] will be extended with the capability to incorporate planning requirements, following the approach described by Ullrich et al. in [17] and [19]. Therefore, the model by Genç will first be introduced in summary before the extensions for the incorporation of planning requirements are described.

3.1. Modeling regularity

To generate regular timetables, the transit network is modeled as a network $N(S, C, t, L, T)$ with a set of stops $S = \{s_1, \dots, s_n\}$, a set of connections $C = \{c_1, \dots, c_v\} \subseteq S \times S$ with travel times $t: C \to \mathbb{N}_0$, and a set of lines $L = \{l_1, \dots, l_m\}$. A line $l_i = (s_{i1}, s_{i2}, \dots, s_{ik(i)})$ is a simple directed path in the directed graph $G = (S, C)$ with a basic interval T_i from the set of all basic intervals $T = \{T_1, \dots, T_m\}$.

For every line l_i the departure time at its first stop s_{i1} is defined by λ_i. A timetable λ from the set of all timetables $\Lambda = \mathbb{N}_{T_1} \times \mathbb{N}_{T_2} \times \dots \times \mathbb{N}_{T_m}$ is a vector with m entries, one for every line. Because only periodic timetables are considered $\mathbb{N}_{T_i} = \{0, \dots, T_i - 1\}$ holds and the departure times at the starting stops can be limited as shown in equation 1.

$$0 \leq \lambda_i \leq T_i - 1 \tag{1}$$

Assuming that the driving times between consecutive stops are fixed and independent of a given timetable, the departure times at all stops $s_{ip} \in S(l_i)$ served by line l_i can be calculated by adding up λ_i and the sum of the travel times for the stops preceding s_{ip} (see equation 2).

$$a(s_{ip}, l_i, \lambda) := \lambda_i + \sum_{k=1}^{p-1} t(s_{ik}, s_{ik+1}) \quad 1 \leq p \leq s_{ik(i)} \tag{2}$$

Under this assumption the departure times at any given stop s are only dependent on the departure times at the starting stops and can also be calculated as shown in equation 3.

$$a(s, l_i, \lambda) = \lambda_i + a(s, l_i, \vec{0}) \tag{3}$$

If it is clear to which timetable a departure time belongs the notation a_i^s is used instead of $a(s, l_i, \lambda)$.

Before formulating the optimization problem different headway measures have to be defined.

Definition 1 (Headway between two lines at a stop)

Let $s \in S$ be a stop of the network served by lines l_i and l_j. Further let λ be the timetable under consideration and let $x_{ij}^s = |a_i^s - a_j^s| mod \gcd(T_i, T_j)$ be the absolute difference between the departure times a_i^s and a_j^s. Then the headway $\delta(s, l_i, l_j, \lambda)$ between l_i and l_j at s und timetable λ can be calculated as follows, as shown by Vince in [21]:

$$\delta(s, l_i, l_j, \lambda) := min\{x_{ij}^s, gcd(T_i, T_j) - x_{ij}^s\} \tag{4}$$

In the best possible case the departure times of two lines with differing basic intervals can be arranged equidistantly in the interval defined by the greatest common divisor of their corresponding basic intervals. Thus the headway can be bound as shown in equation 5.

$$\delta(s, l_i, l_j, \lambda) \leq \left\lfloor \frac{gcd(T_i, T_j)}{2} \right\rfloor \tag{5}$$

Definition 2 (Headway at a stop)
Let $s \in S$ be a stop of the network and let λ be the timetable under consideration. The headway $\delta(s, \lambda)$ at s under λ is the minimum of all headways between all pairs of lines from $L(s)$, the set of lines serving s. It is calculated as shown by equation 6.

$$\delta(s, \lambda) := \begin{cases} \min\limits_{l_i, l_j \in L(s), l_i \neq l_j} \delta(s, l_i, l_j, \lambda), & |L(s)| > 1 \\ T_{L(s)}, & |L(s)| = 1 \end{cases} \tag{6}$$

As a result of equation 5, the upper bound $\bar{\delta}(s)$ for the headway at stop s can be determined by calculating the minimum of all possible equidistant departure distributions (see equation 7).

$$\bar{\delta}(s) = \left\lfloor \min_{l_i, l_j \in L(s), l_i \neq l_j} \frac{\gcd(T_i, T_j)}{2} \right\rfloor \quad L(s) \geq 2 \tag{7}$$

Definition 3 (Headway of a timetable)
Let λ be the timetable for a given network. The headway $\delta(\lambda)$ of this timetable is the minimum of all headways at the stops $s \in S$ of the network and is calculated as shown in equation 8.

$$\delta(\lambda) := \min_{s \in S} \delta(s, \lambda) \tag{8}$$

Definition 4 (Headway sum of a timetable)
Let λ be the timetable for a given network. The headway sum $\delta_\Sigma(\lambda)$ is the sum of all headways at the stops $s \in S$ of the network (see equation 9).

$$\delta_\Sigma(\lambda) := \sum_{s \in S} \delta(s, \lambda) \tag{9}$$

Obviously, an upper bound $\overline{\delta_\Sigma}$ for the headway sum can be calculated by adding up the upper bounds for all stops of the network (see equation 10).

$$\overline{\delta_\Sigma} = \sum_{s \in S} \bar{\delta}(s) \tag{10}$$

With these headway measures the optimization problem to generate regular periodic timetables can be defined.

Definition 5 (Regular periodic timetabling problem)
Let $N(S, C, t, L, T)$ be a network. Find an optimum timetable $\lambda^* \in \Lambda$ with $\delta_\Sigma(\lambda^*) = \delta_\Sigma^*$, where δ_Σ^* is the optimum headway sum defined as shown in equation 11.

$$\delta_\Sigma^* := \max_{\lambda \in \Lambda : \delta(\lambda) = \delta^*} \delta_\Sigma(\lambda) \tag{11}$$

Here δ^* is the optimum timetable headway over all timetables $\lambda \in \Lambda$ (see equation 12).

$$\delta^* := \max_{\lambda \in \Lambda} \delta(\lambda) \tag{12}$$

Timetables solely satisfying equation 12 would only achieve the best possible headway at highly frequented stops, while stops with lower frequencies would be neglected. Timetables satisfying equation 11 on the other hand not only achieve the best possible headway at highly frequented stops, but also consider other stops.

The following mixed integer linear program solves this optimization problem:

$$\max_{\lambda \in \Lambda} \delta(\lambda) * \overline{\delta_\Sigma} + \delta_\Sigma(\lambda) \quad (13)$$

s.t.

$$a_i^s = \lambda_i + a(s, l_i, \vec{0}) \qquad \forall s \in S, l_i \in L(s) \quad (14)$$

$$x_{ij}^s = a_i^s - a_j^s - z_{ij}^s * gcd(T_i, T_j) \qquad \forall s \in S, l_i, l_j \in L(s), l_i \neq l_j \quad (15)$$

$$0 \leq x_{ij}^s \leq gcd(T_i, T_j) - 1 \qquad \forall s \in S, l_i, l_j \in L(s), l_i \neq l_j \quad (16)$$

$$\delta(s, l_i, l_j, \lambda) \leq x_{ij}^s \qquad \forall s \in S, l_i, l_j \in L(s) \quad (17)$$

$$\delta(s, l_i, l_j \lambda) \leq gcd(T_i, T_j) - x_{ij}^s \qquad \forall s \in S, l_i, l_j \in L(s) \quad (18)$$

$$\delta(s, \lambda) \leq \delta(s, l_i, l_j) \qquad \forall s \in S, l_i, l_j \in L(s) \quad (19)$$

$$1 \leq \delta(s, l_i, l_j, \lambda) \leq \left\lfloor \frac{gcd(T_i, T_j)}{2} \right\rfloor \qquad \forall s \in S, l_i, l_j \in L(s) \quad (20)$$

$$1 \leq \delta(\lambda) \leq \delta(s, \lambda) \qquad \forall s \in S \quad (21)$$

$$1 \leq \delta(s, \lambda) \leq \bar{\delta}(s) \qquad \forall s \in S \quad (22)$$

$$0 \leq \lambda_i \leq T_i - 1 \qquad \forall i \in L \quad (23)$$

$$x_{ij}^s, \lambda_i, \in \mathbb{N}_0 \quad (24)$$

$$\delta(s, l_i, l_j, \lambda), \delta(s, \lambda), \delta(\lambda) \in \mathbb{N} \quad (25)$$

$$z_i^s \in \mathbb{Z} \quad (26)$$

Objective function 13 maximizes not only the headway sum but also the timetable headway, making sure every solution satisfies both equation 11 and 12. The timetable headway is weighted by the upper bound of the headway sum $\overline{\delta_\Sigma}$ to assure a balanced consideration of both headway measures during optimization.

Constraint 14 defines the departure times for every line at every stop, which is used by constraint 15 to calculate the distance between pairs of departing lines. Note that the equation used in constraint 15 differs from the equation used in definition 1. Both formulations are equivalent (as shown by Genç in [9]), but the formulation used above does not require the usage of absolute value and modulo operators. Constraint 16 assures the bounds for the distance between departure times. Constraints 17 and 18 make sure equation 4 is adhered to, while constraint 19 assures the same for equation 6. Constraints 20 to 22 assure the headway bounds. All headways have to be at least 1, to assure that no two vehicles depart from the same stop simultaneously. The timetable thus is collision free. Constraint 23 assures that the departure times of all lines stay within their corresponding basic interval. Lastly, constraints 24 to 26 define the domains of all variables.

As can be seen, in order to solve this optimization problem all stops and all possible line pairs at the stops have to be examined, resulting in a high number of variables and inequalities. This number can be reduced substantially using equivalence relations on the set of stops as well as on the edge set of the multi-line conflict graph.

Definition 6 (Multi-line conflict graph)

Given a network $N(S, C, t, L, T)$ the graph $MLCG(N) = (L, M)$ defines the multi-line conflict graph. It has a vertex for every line $l_i \in L$ and edges $(s, l_i, l_j) \in M$ between the vertices of l_i and l_j, if and only if both lines serve stop s, i.e. $(\mathrm{s}, \mathrm{l_i}, \mathrm{l_j}) \in \mathrm{M} \Leftrightarrow \mathrm{s} \in \mathrm{S}(\mathrm{l_i}) \cap \mathrm{S}(\mathrm{l_j})$.

Given the multi-line conflict graph an alternative headway measure can be defined, which will be used to define the equivalence relation on the edges of the multi-line conflict graph.

Definition 7 (Line conflict headway)

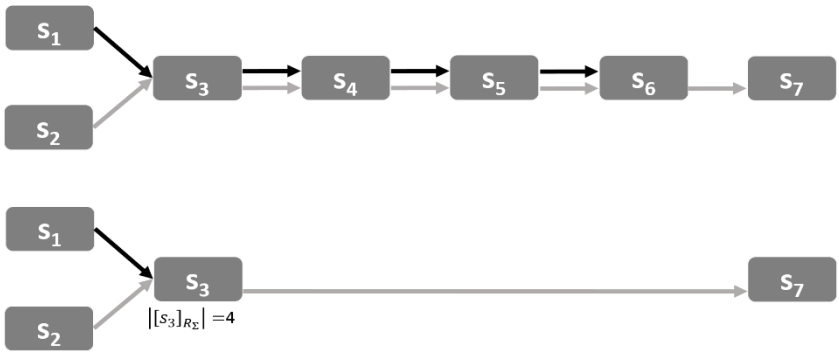

Figure 1: Example of stop reduction for two lines serving four common stops. Stops s_4 s_5, and s_6 can be represented by stop s_3. The equivalence class thus has cardinality four

Let $N(S, C, t, L, T)$ be a network and $MLCG(N) = (L, M)$ the multi-line conflict graph for this network. The headway $\delta(m, \lambda)$ of a conflict $m = (s, l_i, l_j) \in M$ is the headway between the conflicting lines l_i, l_j at stop s and defined as $\delta(\mathrm{m}, \lambda) := \delta(\mathrm{s}, \mathrm{l_i}, \mathrm{l_j})$.

As shown in equation 27, this headway measure can be used to define an alternative way to calculate the headway of a timetable λ instead of equation 8.

$$\delta(\lambda) = \min_{\mathrm{m} \in \mathrm{M}} \delta(\mathrm{m}, \lambda) \tag{27}$$

Definition 8 (Equivalence relation on the edges of the multi-line conflict graph)
Let $F(s_i, s_j, l)$ be the travel time from stop s_i to stop s_j of line l and let M be the edge set of the multi-line conflict graph of the corresponding network. Define a relation R_M on M as follows:

$$\begin{gathered} m_1 = (s_1, l_{11}, l_{12}) \sim_{R_M} m_2 = (s_2, l_{21}, l_{22}) \\ \Leftrightarrow \{l_{11}, l_{12}\} = \{l_{21}, l_{22}\} \wedge F(s_1, s_2, l_{11}) = F(s_1, s_2, l_{12}) \end{gathered} \tag{28}$$

Then the headway for all line conflicts $m_1, m_2 \in M$ for which $m_1 \sim_{R_M} m_2$ holds is equal, i.e. $\forall \lambda \in \Lambda: \delta(\mathrm{m}_1, \lambda) = \delta(\mathrm{m}_2, \lambda)$.

Let $M_0 \subseteq M$ be the set of representatives of M. Then for any given timetable λ of a network equation 29 holds.

$$\forall \lambda \in \Lambda: \min_{\mathrm{m} \in \mathrm{M}} \delta(\mathrm{m}, \lambda) = \min_{\mathrm{m} \in \mathrm{M}_0} \delta(\mathrm{m}, \lambda) \tag{29}$$

As a result, it is sufficient to only examine the set of representatives M_0 when calculating the headway of a timetable.

To further reduce the number of variables and inequalities, another equivalence relation can be defined on the set of stops.

Definition 9 (Equivalence relation on the set of stops)
Define a relation R_Σ on the set of stops S of a given network as follows:

$$\mathrm{s}_1 \sim_{\mathrm{R}_\Sigma} \mathrm{s}_2 \Leftrightarrow \mathrm{L}(\mathrm{s}_1) = \mathrm{L}(\mathrm{s}_2) \wedge \exists \mathrm{k} \in \mathbb{Z} \forall \mathrm{l} \in \mathrm{L}(\mathrm{s}_1): \mathrm{F}(\mathrm{s}_1, \mathrm{s}_2, \mathrm{l}) = \mathrm{k} \tag{30}$$

Under equation 30 two stops $s_i, s_j \in S$ are equivalent to one another, if and only if both are served by the same lines and the travel time between s_i, s_j is equal for all lines. Because travel times are assumed to be fix it is sufficient to only consider the first common stop when calculating the headway in this case and the network can be reduced as shown in figure 1.

Let $S_\Sigma \subseteq S$ be the set of representatives of the factor set S/R_Σ. The headway sum $\delta_\Sigma(\lambda)$ for a given timetable λ can be calculated using the factor set and the cardinalities of the equivalence classes as shown in equation 31.

$$\forall \lambda \in \Lambda: \delta_\Sigma(\lambda) = \sum_{\mathrm{s} \in \mathrm{S}} \delta(\mathrm{s}, \lambda) = \sum_{\mathrm{s} \in \mathrm{S}_\Sigma} \left|[\mathrm{s}]_{\mathrm{R}_\Sigma}\right| * \delta(\mathrm{s}, \lambda) \tag{31}$$

The number of stops that need to be examined to calculate the headway sum can be reduced further, because the headway at stops which are only served by one line is constant as per definition 2.

Let $S_{>1} \subseteq S$ be the set of stops served by more than one line and let $S_{=1} \subseteq S$ be the set of stops served by exactly one line. Then equation 31 can be rewritten as follows:

$$\forall \lambda \in \Lambda: \delta_{\Sigma}(\lambda) = \sum_{s \in S} \delta(s,\lambda) = \sum_{s \in S_{\Sigma,>1}} \left|[s]_{R_{\Sigma}}\right| * \delta(s,\lambda) + \sum_{s \in S_{=1}} T_{L(s)} \tag{32}$$

Because the term $\sum_{s \in S_{=1}} T_{L(s)}$ is constant and independent of the timetable under consideration, it does not have to be examined during the optimization run. Instead it can be added to the objective function value afterwards. The updated mixed integer linear program to generate regular periodic timetables is shown below.

$$\max_{\lambda \in \Lambda} \delta(\lambda) * \overline{\delta_{\Sigma}} + \sum_{s \in S_{\Sigma,>1}} \left|[s]_{R_{\Sigma}}\right| * \delta(s,\lambda) \tag{33}$$

s.t.

$$a_i^s = \lambda_i + a(s, l_i, \vec{0}) \qquad \forall m = (s, l_i, l_j) \in M_0 \tag{34}$$

$$x_{ij}^s = a_i^s - a_j^s - z_{ij}^s * gcd(T_i, T_j) \qquad \forall m = (s, l_i, l_j) \in M_0 \tag{35}$$

$$0 \leq x_{ij}^s \leq gcd(T_i, T_j) - 1 \qquad \forall m = (s, l_i, l_j) \in M_0 \tag{36}$$

$$\delta(m,\lambda) \leq x_{ij}^s \qquad \forall m = (s, l_i, l_j) \in M_0 \tag{37}$$

$$\delta(m,\lambda) \leq gcd(T_i, T_j) - x_{ij}^s \qquad \forall m = (s, l_i, l_j) \in M_0 \tag{38}$$

$$1 \leq \delta(m,\lambda) \leq \left\lfloor \frac{gcd(T_i, T_j)}{2} \right\rfloor \qquad \forall m = (s, l_i, l_j) \in M_0 \tag{39}$$

$$\delta(s,\lambda) \leq \delta(m,\lambda) \qquad \forall s \in S_{\Sigma,>1}, \forall m \in M(s) \tag{40}$$

$$1 \leq \delta(\lambda) \leq \delta(s,\lambda) \qquad \forall s \in S_{\Sigma,>1} \tag{41}$$

$$1 \leq \delta(s,\lambda) \leq \bar{\delta}(s) \qquad \forall s \in S_{\Sigma,>1} \tag{42}$$

$$0 \leq \lambda_i \leq T_i - 1 \qquad \forall i \in L \tag{43}$$

$$x_{ij}^s, \lambda_i, \in \mathbb{N}_0 \tag{44}$$

$$\delta(m,\lambda), \delta(s,\lambda), \delta(\lambda) \in \mathbb{N} \tag{45}$$

$$z_i^s \in \mathbb{Z} \tag{46}$$

The first term of the objective function 33 was not changed compared to objective function 13, the second term was modified according to equations 31 and 32. Constraints 34 to 46 have not changed in context compared to constraints 14 to 26, but now use line conflicts and the reduced stop set instead of all stops and all lines.

3.2. Modeling planning requirements

As explained earlier, while a timetable with optimum regularity can help to raise punctuality and product attractiveness as well as lower peaks in passenger load (see [19], [20]), operational, political, or societal requirements may necessitate deviations in order to make the timetable feasible during daily operation. Ullrich et al. identify five different types of planning requirements that should be considered during optimization:

- Departure times: Define intervals of (in)admissible departure times for some line $l_i \in L$ at a stop $s \in S(l_i)$. This type of requirement can for example be used to incorporate necessary coordination between the departure times of public transit systems with already scheduled departure times of long distance transit networks.
- Headways: Define intervals of (in)admissible spacing between the departures of some line $l_i \in L$ at a stop $s_i \in S(l_i)$ and some other line $l_j \in L, l_j \neq \backslash l_i$ at a stop $s_j \in S(l_j)$.

- Amplifier lines: Trips of a line $l_i \in L$ should be scheduled to relieve trips of a line $l_j \in L, l_j \neq l_i$ in areas with very high demand, by defining intervals of (in)admissible spacing between the departure times at the beginning of a common network section.
- Bi-directional tracks: Define intervals of (in)admissible spacing between the departure times at stops at opposing sides of a bi-directional track to avoid collisions.
- Transfer connections: Define intervals of (in)admissible spacing between the departure times of two lines $l_i, l_j \in L, l_i \neq l_j$ at a stop $s \in S(l_i) \cap S(l_j)$, e.g. to allow passengers of vehicles of the first departing line to transfer to vehicles of the following line.

As can be seen, with the exception of requirements to departure times all requirement types deal with (in)admissible intervals for the spacing between departure times. As a result, Ullrich shows in [16] that (a combination of) the requirement types for departure times and headways can be used to express all other requirement types, subsequently reducing the set of necessary requirement types.

Furthermore, requirements usually can be categorized by some kind of priority system. E.g. while the fulfillment of one requirement could be highly desirable the fulfillment of another requirement might still make the timetable feasible but result in a less desirable configuration.

In order to extend the MILP formulated in section 3.1 with the capability of incorporating prioritized planning requirements the following formalization is employed: Every planning requirement ω in the set Ω of all planning requirements that should be considered during optimization refers to at most two lines $l_i, l_j \in L, l_i \neq l_j$ and two stops $s_i \in S(l_i), s_j \in S(l_j)$, defines exactly one lower bound $\check{t}$ and one upper bound $\bar{t}$ for the corresponding time interval, and holds exactly one priority value ρ from the set of all potential priority values $\mathrm{P} = \{HIGH, MEDIUM, LOW, PROHIBITED\}$. A priority value of $HIGH$ implies a highly desirable requirement, while a value of $PROHIBITED$ signifies an undesirable configuration.

Based on this formalization three different planning requirement types are included into the optimization model, which slightly deviate from the above mentioned requirement types: departure time requirements, headway requirements, and transfer connection requirements. They are formalized as follows:

Definition 10 (Departure time requirement)
Let $\Omega_D \in \Omega$ be the set of all departure time requirements. A member of this set is a 5-tuple $\omega = (l_i, s, \check{t}, \bar{t}, \rho) \in \Omega_D$ defining a lower bound $\check{t} \in \mathbb{N}_{T_i}$ and an upper bound $\bar{t} \in \mathbb{N}_{T_i}$ with $\check{t} \leq \bar{t}$ as well as a priority value $\rho \in \mathrm{P}$ for the departure time a_i^s of line $l_i \in L$ at stop $s \in S(l_i)$ as expressed in equation 47.

$$\check{t} \leq a_i^s - z_i^s * T_i \leq \bar{t} \quad z_i^s \in \mathbb{Z} \tag{47}$$

Definition 11 (Headway requirement)
Let $\Omega_H \in \Omega$ be the set of all headway requirements. A member of this set is a 6-tuple $\omega = (l_i, l_j, s, \check{t}, \bar{t}, \rho) \in \Omega_H$ defining a lower bound $\check{t} \in \{0, \dots, \gcd(T_i, T_j)\}$ and an upper bound $\bar{t} \in \{0, \dots, \gcd(T_i, T_j)\}$ with $\check{t} \leq \bar{t}$ as well as a priority value $\rho \in \mathrm{P}$ for the absolute departure time difference x_{ij}^s of lines $l_i, l_j \in L, i \neq j$ at stop $s \in S(l_i) \cap S(l_j)$ as expressed in equation 48.

$$\check{t} \leq x_{ij}^s \leq \bar{t} \tag{48}$$

Definition 12 (Transfer connection requirement)
Let $\Omega_C \in \Omega$ be the set of all transfer connection requirements. A member of this set is a 7-tuple $\omega = (l_i, l_j, s, r, \check{t}, \bar{t}, \rho) \in \Omega_C$ defining a lower bound $\check{t} \in \{0, \dots, \gcd(T_i, T_j)\}$ and an upper bound $\bar{t} \in \{0, \dots, \gcd(T_i, T_j)\}$ with $\check{t} \leq \bar{t}$ as well as a priority value $\rho \in \mathrm{P}$ for the absolute departure time difference $x_{ij}^{sr} = a_i^s - a_j^r - z_{ij}^{sr} * \gcd(T_i, T_j)$, $z_{ij}^{sr} \in \mathbb{Z}$ of lines $l_i, l_j \in L, l_i \neq l_j$ at stops $s \in S(l_i), r \in S(l_j), s \neq r$ as expressed in equation 49.

$$\check{t} \leq x_{ij}^{sr} \leq \bar{t} \tag{49}$$

Definition 13 (Connected requirements)
Let $\omega_i = (l_i, s_i, \check{t}_i, \bar{t}_i, \rho_i), \omega_j = (l_j, s_j, \check{t}_j, \bar{t}_j, \rho_j) \in \Omega$ be planning requirements of equal type. ω_i and ω_j are said to be connected, if they cover the same lines and the same stops, i.e. $l_i = l_j \wedge s_i = s_j$.

Connected requirements describe logical relationships between (in)admissible intervals on the value range of the corresponding variables and should be considered together during optimization. As an example let $\omega_1, \omega_2, \omega_3$ be three connected departure time requirements, defined as follows: $\omega_1 = (1,1,0,2, HIGH)$, $\omega_2 = (1,1,3,4, MEDIUM)$, and $\omega_3 = (1,1,5,9, PROHIBITED)$.

The requirements describe the following logical relationship between the (in)admissible intervals of the corresponding variables:

$$\left((0 \leq a_1^1 - z_1^1 * T_1 \leq 2) \vee (3 \leq a_1^1 - z_1^1 * T_1 \leq 4)\right) \vee \neg(5 \leq a_1^1 - z_1^1 * T_1 \leq 9)$$

These relationships should be represented in the optimization model. For this purpose, connected requirements with priority values of $HIGH$, $MEDIUM$ or LOW are joined using logical OR, while connected requirements with a priority value of $PROHIBITED$ are joined using logical AND. The resulting partial terms then are combined using a logical OR.

To calculate how well a timetable adheres to a given planning requirement ω, its priority value ρ_ω is associated with a weight $f: \mathrm{P} \to \mathbb{R}$ and the optimization model is extended with a binary variable $b_\omega \in \{0,1\}$ indicating whether or not ω is adhered to or not. The overall adherence of a given timetable λ to a set of planning requirements Ω can then be calculated as shown in equation 50.

$$\Phi(\lambda, \Omega) := \sum_{\omega \in \Omega} \mathrm{b}_\omega * \mathrm{f}(\rho_\omega) \tag{50}$$

Equation 50 cannot simply be added to objective function 35, because their respective value ranges are not comparable. Therefore, equation 52 needs to be normalized. This is done by multiplying it by the ratio β between an upper bound for the regularity value and an upper bound for the requirements adherence (see equation 51).

$$\beta := \frac{\bar{\delta} * \overline{\delta_\Sigma} + \sum_{s \in S_{\Sigma,>1}} \left|[s]_{R_\Sigma}\right| * \bar{\delta}(s)}{|\Omega| * \mathrm{f}(\mathrm{HIGH})} \tag{51}$$

The upper bound for the regularity value can be calculated by inserting the respective upper bounds into the single terms of objective function 33. The upper bound for the requirements adherence can be roughly approximated by multiplying the number of constraints $|\Omega|$ with the theoretically best weight $f(HIGH)$.

The resulting disjunctive program is shown below. Its objective function combines the maximization of regularity with the maximization of normalized requirement adherence and allows to control the importance of those two factors using a weight $0 \leq \alpha \leq 1$.

$$\max_{\lambda \in \Lambda} \ (1-\alpha) * \left(\delta(\lambda) * \overline{\delta_\Sigma} + \sum_{s \in S_{\Sigma,>1}} \left| [s]_{R_\Sigma} \right| * \delta(s,\lambda) \right) + \alpha * \beta * \left(\sum_{\omega \in \Omega} b_\omega * f(\rho_\omega) \right) \tag{52}$$

s.t.

$$a_i^s = \lambda_i + a(s, l_i, \vec{0}) \qquad \forall m = (s, l_i, l_j) \in M_0 \quad (53)$$

$$x_{ij}^s = a_i^s - a_j^s - z_{ij}^s * gcd(T_i, T_j) \qquad \forall m = (s, l_i, l_j) \in M_0 \quad (54)$$

$$0 \le x_{ij}^s \le gcd(T_i, T_j) - 1 \qquad \forall m = (s, l_i, l_j) \in M_0 \quad (55)$$

$$\delta(m, \lambda) \le x_{ij}^s \qquad \forall m = (s, l_i, l_j) \in M_0 \quad (56)$$

$$\delta(m, \lambda) \le gcd(T_i, T_j) - x_{ij}^s \qquad \forall m = (s, l_i, l_j) \in M_0 \quad (57)$$

$$1 \le \delta(m, \lambda) \le \left\lfloor \frac{gcd(T_i, T_j)}{2} \right\rfloor \qquad \forall m = (s, l_i, l_j) \in M_0 \quad (58)$$

$$\delta(s, \lambda) \le \delta(m, \lambda) \qquad \forall s \in S_{\Sigma,>1}, \forall m \in M(s) \quad (59)$$

$$1 \le \delta(\lambda) \le \delta(s, \lambda) \qquad \forall s \in S_{\Sigma,>1} \quad (60)$$

$$1 \le \delta(s, \lambda) \le \bar{\delta}(s) \qquad \forall s \in S_{\Sigma,>1} \quad (61)$$

$$0 \le \lambda_i \le T_i - 1 \qquad \forall i \in L \quad (62)$$

$$\left(\bigvee_{\omega, \rho_\omega \neq PROH.} \check{t} \le a_i^s - z_i^s * T_i \le \bar{t} \right) \bigvee \left(\bigwedge_{\omega, \rho_\omega = PROH.} \neg(\check{t} \le a_i^s - z_i^s * T_i \le \bar{t}) \right) \qquad \forall \omega \ conn., \omega \in \Omega_D \quad (63)$$

$$b_\omega = \begin{cases} 1, & \check{t} \le a_i^s - z_i^s * T_i \le \bar{t} \\ 0, & \neg(\check{t} \le a_i^s - z_i^s * T_i \le \bar{t}) \end{cases} \qquad \forall \omega = (l_i, s, \check{t}, \bar{t}, \rho) \in \Omega_D, \rho \neq PROH. \quad (64)$$

$$b_\omega = \begin{cases} 1, & \neg(\check{t} \le a_i^s - z_i^s * T_i \le \bar{t}) \\ 0, & \check{t} \le a_i^s - z_i^s * T_i \le \bar{t} \end{cases} \qquad \forall \omega = (l_i, s, \check{t}, \bar{t}, \rho) \in \Omega_D, \rho = PROH. \quad (65)$$

$$\left(\bigvee_{\omega, \rho_\omega \neq PROH.} \check{t} \le x_{ij}^s \le \bar{t} \right) \bigvee \left(\bigwedge_{\omega, \rho_\omega = PROH.} \neg(\check{t} \le x_{ij}^s \le \bar{t}) \right) \qquad \forall \omega \ conn., \omega \in \Omega_H \quad (66)$$

$$b_\omega = \begin{cases} 1, & \check{t} \le x_{ij}^s \le \bar{t} \\ 0, & \neg(\check{t} \le x_{ij}^s \le \bar{t}) \end{cases} \qquad \forall \omega = (l_i, l_j, s, \check{t}, \bar{t}, \rho) \in \Omega_H, \rho \neq PROH. \quad (67)$$

$$b_\omega = \begin{cases} 1, & \neg(\check{t} \le x_{ij}^s \le \bar{t}) \\ 0, & \check{t} \le x_{ij}^s \le \bar{t} \end{cases} \qquad \forall \omega = (l_i, l_j, s, \check{t}, \bar{t}, \rho) \in \Omega_H, \rho = PROH. \quad (68)$$

$$x_{ij}^{sr} = a_i^s - a_j^r - z_{ij}^{sr} * gcd(T_i, T_j) \qquad \forall \omega = (l_i, l_j, s, r, \check{t}, \bar{t}, \rho) \in \Omega_C \quad (69)$$

$$0 \le x_{ij}^{sr} \le gcd(T_i, T_j) - 1 \qquad \forall \omega = (l_i, l_j, s, r, \check{t}, \bar{t}, \rho) \in \Omega_C \quad (70)$$

$$\left(\bigvee_{\omega, \rho_\omega \neq PROH.} \check{t} \le x_{ij}^{sr} \le \bar{t} \right) \bigvee \left(\bigwedge_{\omega, \rho_\omega = PROH.} \neg(\check{t} \le x_{ij}^{sr} \le \bar{t}) \right) \qquad \forall \omega conn., \omega \in \Omega_C \quad (71)$$

$$b_\omega = \begin{cases} 1, & \check{t} \le x_{ij}^{sr} \le \bar{t} \\ 0, & \neg(\check{t} \le x_{ij}^{sr} \le \bar{t}) \end{cases} \qquad \forall \omega = (l_i, l_j, s, r, \check{t}, \bar{t}, \rho) \in \Omega_C, \rho \neq PROH. \quad (72)$$

$$b_\omega = \begin{cases} 1, & \neg(\check{t} \le x_{ij}^{sr} \le \bar{t}) \\ 0, & \check{t} \le x_{ij}^{sr} \le \bar{t} \end{cases} \qquad \forall \omega = (l_i, l_j, s, r, \check{t}, \bar{t}, \rho) \in \Omega_C, \rho = PROH. \quad (73)$$

$$\beta = \frac{\bar{\delta} * \overline{\delta_\Sigma} + \sum_{s \in S_{\Sigma,>1}} \left| [s]_{R_\Sigma} \right| * \bar{\delta}(s)}{|\Omega| * f(HIGH)} \tag{74}$$

$$x_{ij}^s, x_{ij}^{sr}, \lambda_i, \in \mathbb{N}_0 \quad (75)$$

$$\delta(m, \lambda), \delta(s, \lambda), \delta(\lambda) \in \mathbb{N} \quad (76)$$

$$z_i^s, z_{ij}^s \in \mathbb{Z} \quad (77)$$

$$b_\omega \in \{0,1\} \quad (78)$$

$$0 \le \alpha \le 1 \quad (79)$$

Constraints 53 to 62 handle arrival times, spacing between consecutive arrivals, headways, and departure times at the lines' first stops as seen before in section 3.1 (see constraints 34 to 43). Constraints 63 to 65 make sure the arrival times adhere to the bounds defined by the given departure requirements, while constraints 66 to 68 assure the same thing for the given headway requirements. Constraints 69 and 70 define the variables and domains for the departure spacing between lines at separate stops as discussed in definition 12, while constraints 71 to 73 assure the respective variables adhere to the bounds specified by the given transfer connection requirements. Lastly, constraint 74 defines the normalizing factor and constraints 75 to 79 establish the domains of the variables.

3.3. Complexity of the optimization model

The complexity of the proposed optimization model depends on the number of line conflicts $|M_0|$, the number of stops served by more than one line $|S_{\Sigma,>1}|$, the number of lines $|L|$ and their basic intervals T, and the number of planning requirements $|\Omega_D|$, $|\Omega_H|$, and $|\Omega_C|$. The optimization problem for a given transit network contains at least $6 * |M_0| + 3 * |S_{\Sigma,>1}| + |L| + 2 * |\Omega_D| + 2 * |\Omega_H| + 4 * |\Omega_C| + 1$ constraints and $5 * |M_0| + |S_{\Sigma,>1}| + |L| + 2 * |\Omega_D| + |\Omega_H| + 4 * |\Omega_C| + 1$ variables. The solution space of the optimization problem for a given transit network contains $\prod_{l \in L} T_l$ potential solutions in the worst case.

Because of the potentially very large solution space the run time required to solve the proposed model to optimality for network instances of realistic size can be quite high, as will be seen in the next chapter. However, as shown by Genç in [9] the run time can be reduced substantially, if the graph representing the network under consideration consists of multiple connected components, which are physically independent from each other. Because physically independent connected components do not share any lines or stops, a simple divide and conquer approach can be utilized, i.e. for every component an individual timetable can be constructed by solving a reduced optimization problem containing only the lines and stops belonging to the specific component. Afterwards, the partial timetables can be consolidated into an overall timetable for the whole network.

It has to be noted that the divide and conquer approach cannot be utilized if lines and stops of different connected components are linked via planning requirements (e.g. a line of component A has to depart after a line of component B). In this case, while the connected components remain physically independent from each other, they are logically linked via the planning requirement and subsequently need to be considered simultaneously during optimization.

4. Experiments

To validate the proposed model, twelve-hour-timetables from 7 am to 7 pm for models of three transit networks with differing sizes and characteristics (see table 1) are generated using a self-developed Java application which applies the IBM ILOG CPLEX 12.5 Java library to solve the corresponding optimization problems. The computer system used is an AMD Athlon 64 X2 Dual Core processor with 2.10 GHz and 4 GB RAM. If planning requirements are included, the following weights are associated with the priority values: $f(HIGH) = 1, f(MEDIUM) = 0.25, f(LOW) = 0.125, f(PROHIBITED) = 0.03125$.

Network	$\|L\|$	$\|M\|$	$\|M_0\|$	$\|S\|$	$\|S_{\Sigma,>1}\|$
LCL	6	50	6	20	4
UCL	8	64	17	40	13
Cologne	70	313	179	1242	126

Table 1: Characteristics of the examined transit networks

4.1. Experiments on models of artificial networks

In a first set of experiments the correctness of the optimization model and the permissibility of the generated timetables are verified by applying it to models of two artificial networks, called Linear City Link (LCL) and Universal City Link (UCL).

4.1.1 Linear City Link

LCL is an artificial network with a very simple structure, allowing for easy manual verification of the obtained results (see figure 2). Timetables with a common basic interval of ten minutes are generated.

Without applying planning requirements an optimum regular timetable λ^* for LCL can achieve a headway of $\bar{\delta}(\lambda^*) = 3$ minutes in the best case, arranging the departure times of lines 1-B01, 2-B01, 3-B01 and 1-B02, 2-B02, 3-B02 at their common stops in such a way that their headways correspond to a permutation of the tuple (3,3,4). The objective function of such a timetable has the following upper bound:

$$\underbrace{(3 * 78)}_{\bar{\delta}(\lambda^*)*\overline{\delta_\Sigma}} + \underbrace{(1 * 5 + 1 * 5 + 3 * 8 + 3 * 8)}_{\sum_{s \in S_{\Sigma,>1}} |[s]\_R_\Sigma| * \bar{\delta}(s)} + \underbrace{(10 + 10)}_{\sum_{s \in S_{=1}} \bar{\delta}(s)} = 312$$

The optimum timetable computed by our application achieves an objective function value of 310 and a timetable headway of three minutes. The deviation of one percent between the computed objective function value and its upper bound can be explained by examining the calculation of the upper bound for the stop headway in more detail: When calculating the upper bound for the stop headway every stop is considered isolated. This has no consequences as long as the neighboring stops are served by the same lines, keeping to the same headways. If this condition is not met, one of the stops will not be able to achieve its theoretically optimum headway and subsequently the objective function value will not reach its upper bound. As an example consider stops 2 and 3 of network LCL. At stop 2 only lines 1-B01 and 2-B01 depart, resulting in an upper bound for the stop headway of $\bar{\delta}(2) = 5$ minutes. At stop 3 line 3-B01 joins lines 1-B01 and 2-B01, thus the upper bound for the headway at this stop is $\bar{\delta}(3) = 3$ minutes. Because the majority of stops (i.e. 2 to 10 and 20 to 13) of LCL are served by the same three lines the optimization will prioritize achieving the best possible headway at these stops. Thus, it arranges the departures of lines 1-B01 and 2-B01 with a headway of four minutes at stop 3, which subsequently results in a headway of four minutes at stop 2 instead of the theoretically optimum of five minutes. The same is true for lines 1-B02 and 2-B02 at stop 12.

To examine the impacts of planning requirements two headway requirements $\omega_1 = (1 - B01{,}2 - B01{,}2{,}5{,}5, HIGH)$ and $\omega_2 = (1 - B02{,}2 - B02{,}12{,}5{,}5, HIGH)$ are defined, enforcing the aforementioned theoretically optimal headways of five minutes between lines 1-B01 and 2-B01 at stop 2 and lines 1-B02 and 2-B02 at stop 12, respectively. With a weight of $\alpha = 0.5$ the optimization generates a timetable with an objective function value of $0.5 * 218 + 0.5 * 146 * 2 = 255$ and a timetable headway of two minutes. As expected, the introduction of the planning requirements results in a loss in regularity due to the non-optimal distribution of departure times at stops 3 to 10 and 13 to 20.

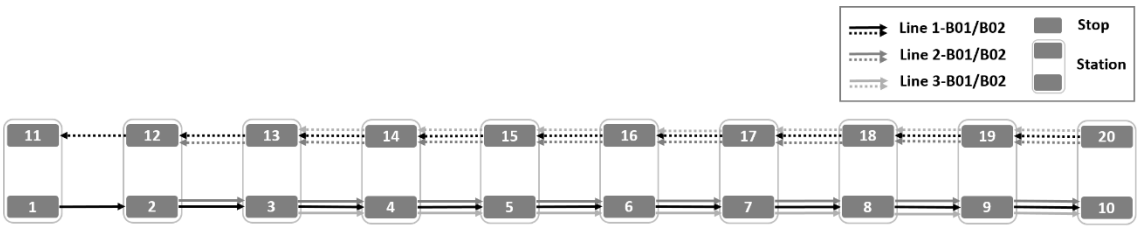

Figure 2: Artificial transit network LCL

4.1.2 Universal City Link

For a second set of experiments the artificial network UCL is employed (see figure 3). Contrary to LCL it has a more realistic, i.e. irregular, network structure and with circle lines 2-B01 and 2-B02 includes two special line structures. To verify the correctness of the optimization model under varying conditions differing basic intervals for the different lines are employed: lines 1-B01, 1-B02, 3-B01, 3-B02, 4-B01, and 4-B02 receive a basic interval of ten minutes, while lines 2-B01 and 2-B02 get a basic interval of 20 minutes.

Without applying planning requirements an optimum regular timetable λ^* for UCL can achieve a headway of $\bar{\delta}(\lambda^*) = 3$ minutes at best. The objective function value of such a timetable then has an upper bound of 848.

The optimum timetable computed using the optimization achieves an objective function value of 836 and a timetable headway of three minutes. Once again the difference of two percent between the achieved objective function value and the upper bound can be attributed to the isolated examination of stops during the calculation of the upper bounds. E.g. stop 1141 has an upper bound for the stop headway of five minutes, but the optimum timetable, which also considers departures at neighboring stops, can only achieve a headway of four minutes in the best case.

To examine a more realistic scenario, eight exemplary transfer connections are modeled (see table 2, columns 2 to 4).

No.	From	At	To	At	Priority			
					HIGH	MEDIUM	LOW	PROHIBITED
1	2-B01	1143	1-B01	1141	3 4	5	2 6 7	0 1 8 9
2	2-B01	1142	4-B01	1141	3 4	5	2 6 7	0 1 8 9
3	3-B01	1043	2-B01	1041	3 4	5	2 6 7	0 1 8 9
4	3-B02	1143	1-B01	1141	3 4	5	2 6 7	0 1 8 9
5	3-B02	1042	2-B01	1041	3 4	5	2 6 7	0 1 8 9
6	4-B01	1041	2-B02	1043	3 4	5	2 6 7	0 1 8 9
7	4-B02	1142	2-B01	1143	3 4	5	2 6 7	0 1 8 9
8	4-B02	1042	2-B02	1043	3 4	5	2 6 7	0 1 8 9

Table 2: Exemplary transfer connections between lines from one stop to another, UCL

The specific requirements defined should result in transfer connections with a low waiting time, but not so low that passengers may narrowly miss their connection. Thus transfer connections of three to four minutes are valued the most (see table 2, column 5), followed by transfer connections of five minutes (see table 2, column 6). Transfer connections of two minutes as well as six to seven minutes are still tolerable (see table 2, column 7), while transfer connections of zero to one and eight to nine minutes have to be avoided (see table 2, column 8). This results in the 48 constraints visible in Table 2, i.e. six connected requirements for each transfer connection to be modeled.

With weight $\alpha = 0.5$ the optimization computes an optimum timetable with an objective function value of $0.5 * 600 + 0.5 * 16 * 5.875 = 347$ and a timetable headway of two minutes. As shown by table 3, the optimization assigns highly prioritized values for five of the eight defined transfer connections. Only the transfer connections from lines 2-B01 to 4-B01, 3-B02 to 1-B01, and 4-B01 to 2-B02 are assigned low priority values.

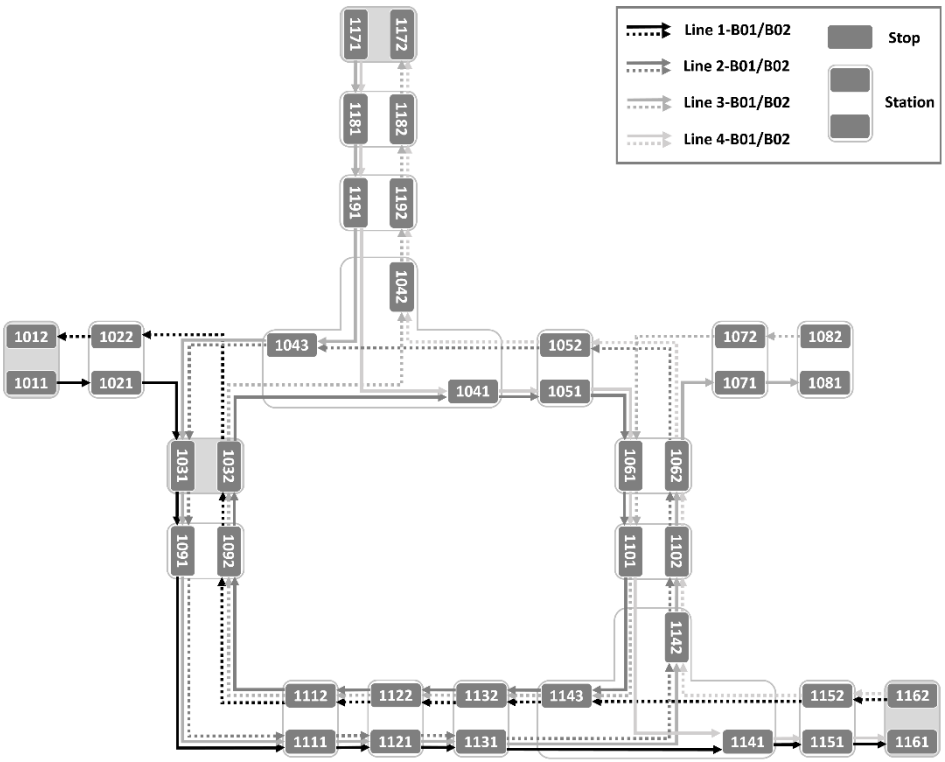

Figure 3: Artificial transit network UCL

No.	From	At	To	At	Headway	Priority
1	2-B01	1143	1-B01	1141	4	HIGH
2	2-B01	1142	4-B01	1141	2	LOW
3	3-B01	1043	2-B01	1041	3	HIGH
4	3-B02	1143	1-B01	1141	7	LOW
5	3-B02	1042	2-B01	1041	3	HIGH
6	4-B01	1041	2-B02	1043	2	LOW
7	4-B02	1142	2-B01	1143	3	HIGH
8	4-B02	1042	2-B02	1043	3	HIGH

Table 3: Achieved time intervals for connections between lines, UCL

4.2. Experiments on Cologne's bus network

To examine the efficiency of the proposed approach under realistic conditions the optimization is applied to a model based on data of the bus network of Cologne from 2001. Timetables with a common basic interval of ten minutes and no planning requirements are generated.

Under these conditions an optimum regular timetable λ^* for the network of Cologne can at best achieve a headway of $\bar{\delta}(\lambda^*) = 2$ minutes and an objective function value of 33,729. The optimum timetable computed by CPLEX achieves an objective function value of 33,509 (a one percent deviation from the upper bound) with a headway of two minutes. Table 4 compares the number of stops with a specific stop headway under the optimum timetable generated using the optimization with the maximum possible numbers if all stops would assume their respective upper bound. Because of the isolated consideration of stops when calculating the upper bounds, the generated timetable deviates from the theoretically possible numbers. However, the optimization is able to reach the upper bound for at least 116 (or 55 percent) of the 210 stops served by more than one line.

Headway	1	2	3	4	5	6	7	8	9	10
Maximum number	0	13	19	0	178	0	0	0	0	1027
Optimum timetable	0	36	38	33	103	0	0	0	0	1027

Table 4: Number of stops with certain headways, Cologne

4.3. Comparison and discussions

As discussed earlier, the efficiency of the proposed approach depends highly on the size and structure of the transit network under consideration, as can be seen in table 5. It shows average run times over ten optimization runs for the three examined transit networks split into the three distinct phases of the optimization: preprocessing the network and constraint information into a solvable CPLEX model, solving this model with CPLEX, and translating the results from CPLEX into a set of actual trips with distinct starting times which can be stored in a database.

Network	Preprocessing [Sec.]	CPLEX [Sec.]	Postprocessing [Sec.]	Overall runtime [Sec.]	Gap [%]
LCL	0.90	0.59	18.71	20.20	-
UCL	0.90	0.78	22.24	23.92	-
Cologne	7.09	> 8,861.66	185.16	> 9,053.91	0.3

Table 5: Average times for different phases of the optimization

The time spent in the preprocessing phase increases with the number of stops and lines, because the identification of line conflicts and equivalence classes gets more complex. The time spent in postprocessing increases with the number of lines as well as the length of the planning period, as generating timetables for fewer lines and shorter periods results in the generation of fewer trips and subsequently requires fewer time consuming database operations. For LCL and UCL the time spent in postprocessing is especially significant as it represents 93 percent of the overall run time. The actual optimization problem is solved virtually instantaneously for LCL and UCL, but requires more than 2.5 hours for the network of Cologne, due to its size. Therefore, a gap, i.e. a relative tolerance between the best integer objective and the upper bound calculated by CPLEX, of 0.3 percent was applied in order to obtain solutions in acceptable time. Without the specification of a gap the algorithm was not able to finish in acceptable time.

Nonetheless, the timetable discussed in section 4.2 is an optimum regular timetable for the network of Cologne, because fortunately the data for the bus network of Cologne results in a graph consisting of five physically independent connected components, enabling the application of the divide and conquer approach described in section 3.3. To ensure the optimality of the thusly generated timetable for the network of Cologne, the solution obtained by CPLEX leveraging the separation approach was afterwards used as start configuration for an optimization run without separation, resulting in no further improvement and thus proving the optimality.

How the separation into connected components affects the run time can be seen in table 6. While network UCL cannot be decomposed (and subsequently has no entry in table 6), network LCL, which can be decomposed, does not profit from the approach and actually yields an insignificantly higher run time. The network of Cologne on the other hand profits significantly from the described approach, reducing the runtime by at least 82 percent to about 26 minutes.

Network	Preprocessing [Sec.]	CPLEX [Sec.]	Postprocessing [Sec.]	Overall runtime [Sec.]
LCL	0.86	0.73	18.88	20.47
Cologne	7.36	1,577.68	186.49	1,771.53

Table 6: Average time of different phases of the optimization with separation

5. Conclusions

In this paper a disjunctive program formulation for the generation of regular transit timetables, which also adhere to prioritized planning requirements was proposed. These requirements include fixed departure times, headways at common stops, and transfer connections.

The approach was applied to models of three different transit networks: two small artificial networks, as well as a model of the bus network of Cologne, Germany. The conducted experiments showed that the proposed approach can indeed be used to generate regular timetables, which will adhere to given sets of planning requirements, although fulfilling given planning requirements may result in lower regularity.

Furthermore, it was shown that the size and structure of the examined transit network heavily influences the run time, but if the transit network can be separated into physically independent connected components, a simple divide and conquer approach can be used to significantly reduce run time, by separately optimizing the connected components and merging the resulting partial timetables into an overall timetable.

By including planning requirements during the planning phase, fewer manual adjustments are needed to assure feasibility of the generated timetables during daily operation. Although this improvement of practicability is important, it can only be a first step. To further ensure the practicability of the generated timetables, they have to be exposed to dynamic conditions, e.g. variations in travel and boarding times. One way to achieve this goal without costly trial and error during actual operations is to employ simulation methods. For this reason, a new discrete event based simulation framework will be developed allowing to evaluate the practicability of generated timetables under dynamic conditions.

Acknowledgements

This material is based in part upon work supported by the National Science Foundation under Grant Nos. I/UCRC IIP-1338922, AIR IIP-1237818, III-Large IIS-1213026, MRI CNS-1532061, MRI CNS-1429345, MRI CNS-0821345, MRI CNS-1126619, MRI CNS-0959985, RAPID CNS-1507611, and U.S. DOT Grant ARI73

References

[1] E. Bampas, G. Kaouri, M. Lampis, A. and Pagourtzis, Periodic Metro Scheduling, Proc 6th Workshop on ATMOS, Dagstuhl, Germany, 2006, pp. 63-77.

[2] N. Bešinović, R. M. P. Goverde, E. Quaglietta, and R. Roberti, An integrated micro-macro approach to robust railway timetabling, Transport Red B-Meth 87 (2016), pp. 14-32.

[3] V. Cacchiani, A. Caprara, and M. Fischetti, A Lagrangian Heuristic for Robustness with an Application to Train Timetabling, Transport Sci 46 (2012), pp. 124-133.

[4] A. Ceder, B. Golany, and O. Tal, Creating bus timetables with maximal synchronization, Transport Res A-Pol 35 (2001), pp. 913-928.

[5] R. J. Dakin, A tree-search algorithm for mixed integer programming problems, Comput J 8 (1965), pp. 250-255.

[6] G. Desaulniers and M. Hickman, “Public Transit”, Handbooks in Operations Research and management Science, C. Barnhart and G. Laporte (Editors), North Holland, Amsterdam, 2007, pp. 69-127.

[7] J. Dreo, A. Petrowski, A. Siarry, and E. Taillard, Metaheuristics for Hard Optimization, Springer, Berlin Heidelberg, 2006.

[8] A. Eranki, A model to create bus timetables to attain maximum synchronization considering waiting times at transfer stops, Master thesis, University of South Florida, Tampa, 2004.

[9] Z. Genç, Ein neuer Ansatz zur Fahrplanoptimierung im ÖPNV: Maximierung von zeitlichen Sicherheitsabständen, Dissertation, University of Cologne, Germany, 2003.
[10] R. M. P. Goverde, N. Bešinović, A. Binder, V. Cacchiani, R. Roberti, and P. Toth, A three-level framrwork for performance-based railway timetabling, Transport Res C-Emer 67 (2016), pp. 62-83.
[11] O. J. Ibarra-Rojas and Y. A. Rios-Solis, Synchronization of bus timetabling, Transport Res B-Meth 46 (2012), pp. 599-614.
[12] O. J. Ibarra-Rojas, F. López-Irarragorri, and Y. A. Rios-Solis, Multiperiod synchronization bus timetabling, Transport Sci 50 (2015), pp. 805-822.
[13] C. Liebchen, M. Schachtebeck, A. Schöbel, S. Stiller, and A. Prigge, Computing delay resistant railway timetables, Comput Opers Res 37 (2010), pp. 857-868.
[14] G. K. D. Saharidis, C. Dimitropoulos, and E. Skordilis, Minimizing waiting times at transitional nodes for public bus transportation in Greece, Oper Res 14 (2014), pp. 341-359.
[15] M. A. Salido and A. Giret, Feasible distributed CSP models for scheduling problems, Eng Appl Artif Intel 21 (2008), pp. 723-732.
[16] O. Ullrich, Modellbasierte Parallelisierung von Anwendungen zur Verkehrssimulation - Ein dynamischer und adaptiver Ansatz, Dissertation, University of Cologne, Germany, 2014.
[17] O. Ullrich, D. Lückerath, S. Franz, and E. Speckenmeyer, Simulation and optimization of Cologne's tram schedule, Simulation Notes Europe 22 (2012), pp. 69-76.
[18] O. Ullrich, D. Lückerath, and E. Speckenmeyer, A robust schedule for Montpellier's Tramway network, Simulation Notes Europe,25 (2015), pp. 1-8.
[19] O. Ullrich, D. Lückerath, and E. Speckenmeyer, Do regular timetables help to reduce delays in tram networks? – It depends!, Public Transport 8 (2016), pp. 39-56.
[20] N. van Oort and R. van Nes, Regularity analysis for optimizing urban transit network design, Public Transport 1 (2009), pp. 155-168.
[21] A. Vince, Scheduling periodic events, Discr Appl Math 25 (1989), pp. 2999-310.
[22] Y. Wu, J. Tang, Y. Yu, and Z. Pan, A stochastic optimization model for transit network timetable design to mitigate the randomness of traveling time by adding slack time, Transport Res C-Emer 52 (2015), pp. 15-31.
[23] Y. Yan, Q. Meng, S. Wang, and X. Guo, Robust optimization model of schedule design for a fixed bus route, Transport Res C-Emer, 25 (2012), pp. 113-121.