Available online at www.sciencedirect.com
ScienceDirect
European Journal of Operational Research 193 (2009) 867–884
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Solving a school bus scheduling problem with integer programming

Armin Fügenschuh

Darmstadt University of Technology, Schlossgartenstrasse 7, 64289 Darmstadt, Germany

Received 10 November 2006; accepted 15 October 2007
Available online 12 November 2007

Abstract

In many rural areas in Germany pupils on the way to school are a large if not the largest group of customers in public transport. If all schools start more or less at the same time then the bus companies need a high number of vehicles to serve the customer peak in the morning rush hours. In this article, we present an integer programming model for the integrated coordination of the school starting times and the public bus services. We discuss preprocessing techniques, model reformulations, and cutting planes that can be incorporated into a branch-and-cut algorithm. Computational results show that in our test counties a much lower number of buses would be sufficient if the schools start at different times.
© 2007 Elsevier B.V. All rights reserved.

Keywords: Integer programming; Vehicle scheduling; Coupled time windows; Real-world applications

1. Introduction

Traffic peaks are peaks in cost. This is in particular true for rural counties, where public mass transportation is focused on the demand of pupils. About half to two-third of pupils in rural areas in Germany take a bus to get to school. Most of them are integrated in the public bus system, a minority is transferred by special purpose school buses. In both cases the respective county in which the pupils live is responsible for their transfer to school. This in particular means that the county administration pays the fees. Since tax money is a scarce resource in our days, the administration has great interest in reducing their payments as much as possible.

It was noted by Stöveken [19] that a significant lower number of buses is needed, if the bus scheduling problem is solved together with the starting time problem, i.e., the simultaneous settlement of school and trip starting times. A small intuitive example is shown in Fig. 1. If two schools start at the same time then two buses are necessary to bring the pupils to their respective schools. If they start at different times then one bus is sufficient to bring the pupils to both schools.

This article deals with the question how to roll out this intuitive example to a real-world situation, where we have to deal with bus companies operating up to several hundred buses in counties having up to 100 schools, and 15,000 pupils waiting every morning at 3000 bus stops. In our application, the costs are proportional to the number of buses in use. Of course, if fewer buses are deployed, each single bus has a higher driving load, which then leads to higher operating and maintenance costs per bus. However, it is safe to assume that those costs are marginal compared to the costs for keeping too much buses in stock.

A wide range of transportation problems involving public bus transit and the organization of transporting pupils to schools were already studied before, for example, see [14,3,5,4,8,6,16]. However, none of the presented models completely fits to our problem because of the following reasons: In all these approaches the school starting times are fixed and cannot be changed to save buses. Moreover pupils are always transported directly to school, and changing the bus is not allowed.

E-mail address: fuegenschuh@mathematik.tu-darmstadt.de

0377-2217/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2007.10.055

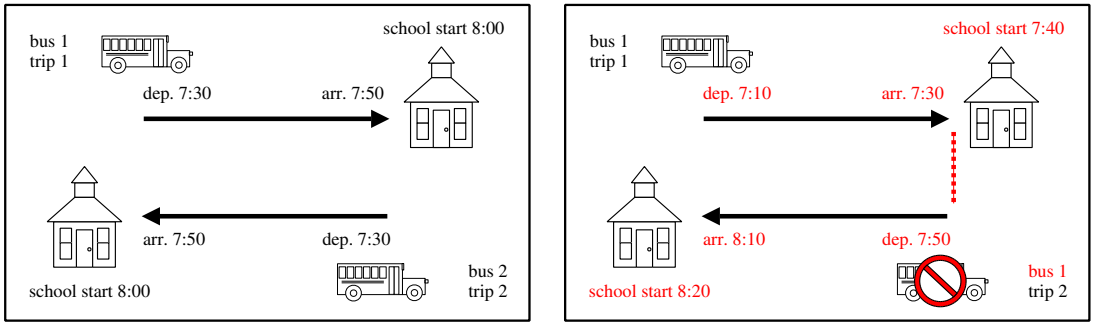

Fig. 1. The central idea (before–after).

Locating bus stops, designing routes (trips) and assigning pupils to routes is part of some of the optimization models, where in our case these are input figures. Finally, scheduling drivers is not issued: Since our time horizon is small (from 5:00 to 9:00 a.m.), planning of breaks is not needed. In the next section, we present a new model that is tailored for our problem settings.

It will turn out that this new model belongs to the class of vehicle routing (or vehicle scheduling) problems with time windows (VRPTW). The new aspect are coupling conditions among the time windows. Since VRPTW belongs to the class of NP-hard problems, we cannot expect polynomial time algorithms for their solution (unless $P = \mathrm{NP}$). Hence one approach which we present in this article is the use of a linear programming based branch-and-cut algorithm for the numerical solution of our instances.

2. An integer programming model

Our model is formulated as an integer programming problem based on the vehicle routing problem with time windows (VRPTW). For details on the VRPTW, we refer to Cordeau et al. [7].

2.1. Sets and parameters

We assume that it is desired to study the integrated planning problem in a given area or region, and at a certain period of the day. In the sequel, we call this area a *county*, but it might also correspond to less than a county (such as a single city) or more (for example if a large bus company operates in several adjacent counties). The period of the day we are mainly interested in are the morning rush hours (which gives a time horizon from approximately 5:00 to 9:00 a.m.), so in general we only consider the school starting times (and not, for instance, school ending times). We start with a detailed description of the necessary input sets and parameters of the model, see Table 1 for an overview.

2.1.1. Trips

Let $\mathscr{V}$ be the set of all passenger trips in the given county. A *passenger trip* (also *bus trip*, or *trip* for short) $t \in \mathscr{V}$ is a sequence of bus stops. To each bus stop an arrival and a departure time (of some bus) is assigned. The time difference between the departure at the first and the arrival at the last bus stop is called the *service duration*, and is denoted by $\delta_t^{\mathrm{trip}} \in \mathbb{Z}_+$. (Generally all time-related parameters and variables in the model are integral with the unit "minute".) The *current starting time* of trip t, i.e., the departure time of a bus at the first bus stop in t, is given by $\hat{\alpha}_t \in \mathbb{Z}_+$. We assume that a time window $\underline{\alpha}_t, \bar{\alpha}_t \in \mathbb{Z}_+$, $\underline{\alpha}_t \leqslant \bar{\alpha}_t$ is given, in which the *planned starting time* must lie. Note that once the starting time of the trip is settled, the departure and arrival times at all other bus stops within this trip are also determined. We distinguish the following types of trips in $\mathscr{V}$: School trips, feeder and collector trips, and free trips. Their definitions are given below.

The trips are served by vehicles (buses). The buses start and end their services at the *depot*. The trip without passengers from the depot to the first bus stop of trip t is called *pull-out trip*. Denote $\delta_t^{\mathrm{out}} \in \mathbb{Z}_+$ the time a bus needs to drive this pull-out trip. When the bus arrives at the last bus stop of trip t, it is either sent on the *pull-in trip*, i.e., back to the depot, or it is re-used to serve another trip. The duration of the pull-in trip is denoted by $\delta_t^{\mathrm{in}} \in \mathbb{Z}_+$.

Instead of sending the bus back to the depot after having served a trip it is of course more sensible to re-use the bus to serve other trips, as long as this is possible. Thus, we seek such a connection of trips. Let the set $\mathscr{A} \subset \mathscr{V} \times \mathscr{V}$ contain all pairs of trips (t_1, t_2) that can be connected. The intermediate trip from the last bus stop of trip t_1 to the first bus stop of trip t_2, where no passengers are transported, is called a *shift* or a *deadhead trip*. The duration of the deadhead trip is given by $\delta_{t_1 t_2}^{\mathrm{shift}} \in \mathbb{Z}_+$. In order to absorb possible delays (due to traffic jams, for example) a minimum waiting time $\underline{\omega}_{t_1 t_2}^{\mathrm{idle}} \in \mathbb{Z}_+$, typically 2 minutes, can be imposed after the deadhead trip before the next passenger trip starts. A maximum waiting time

Table 1
Input data: Sets and parameters

$\mathscr{V} \ni t$	Bus trips (nodes)
$\mathscr{A} \ni (t_1, t_2)$	Connectable trips (arcs)
$\mathscr{S} \ni s$	Schools
$\mathscr{P} \ni (s, t)$	School-trip pairings
$\mathscr{C} \ni (t_1, t_2)$	Feeder and collector trip pairings
$\hat{\tau}_s$	Current school starting time
$\hat{\alpha}_t$	Current trip starting time
δ_t^{trip}	Time for serving entire trip
$\delta_{st}^{\text{school}}$	Time for serving trip from start to school
$\delta_{t_1t_2}^{\text{feeder}}$	Time for feeder trip from start to changing bus stop
$\delta_{t_1t_2}^{\text{collector}}$	Time for collector trip from start to changing bus stop
δ_t^{out}	Time for pull-out
δ_t^{in}	Time for pull-in trip
$\delta_{t_1t_2}^{\text{shift}}$	Time for deadhead trip
$\underline{\tau}_s, \bar{\tau}_s$	Bounds on school starting time (lower, upper)
$\underline{\alpha}_t, \bar{\alpha}_t$	Bounds on trip starting time
$\underline{\omega}_{st}^{\text{school}}, \bar{\omega}_{st}^{\text{school}}$	Bounds on waiting time for pupils at school
$\underline{\omega}_{t_1t_2}^{\text{change}}, \bar{\omega}_{t_1t_2}^{\text{change}}$	Bounds on waiting time at changing bus stop
$\underline{\omega}_{t_1t_2}^{\text{idle}}, \bar{\omega}_{t_1t_2}^{\text{idle}}$	Bounds on waiting time after deadhead trip

$\bar{\omega}_{t_1t_2}^{\text{idle}} \in \mathbb{Z}_+$, $\underline{\omega}_{t_1t_2}^{\text{idle}} \leqslant \bar{\omega}_{t_1t_2}^{\text{idle}}$, can also be specified if the bus company wants to avoid long idle times of their buses. A typical value used in practice is 30 minutes.

The connection of a pull-out trip, several passenger and intermediate deadhead trips and a final pull-in trip which are to be served by one and the same bus is called a *block* or *schedule* (see Fig. 2). One can think of $(\mathscr{V}, \mathscr{A})$ as a directed graph, where the node set corresponds to the passenger trips and the arc set corresponds to the deadhead trips. Then a block can be interpreted as a path in this graph.

2.1.2. Schools

Let $\mathscr{S}$ be the set of all schools in the county under consideration. The *current starting time* of school $s \in \mathscr{S}$ is given by $\hat{\tau}_s \in \mathbb{Z}_+$. It is allowed to change this starting time within some time window $\underline{\tau}_s, \bar{\tau}_s \in \mathbb{Z}_+$, $\underline{\tau}_s \leqslant \bar{\tau}_s$. Usually, this time window reflects the legal bounds on the school starting time (from 7:30 to 8:30 a.m.). Each school has one or more bus stops, where pupils get off the bus and walk the rest of the way to school.

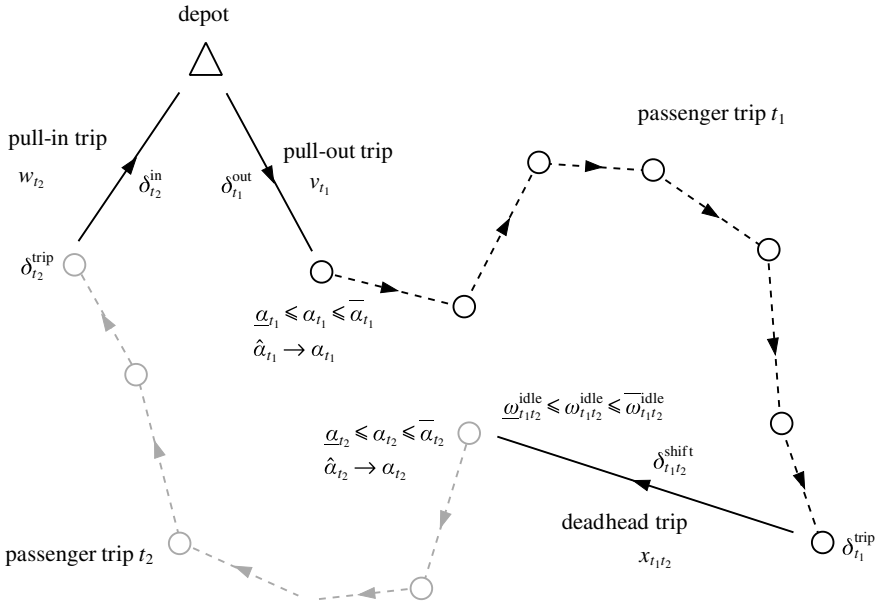

Fig. 2. Two trips t_1, t_2 in a schedule.

The set $\mathscr{P} \subset \mathscr{S} \times \mathscr{V}$ consists of pairs (s,t), where trip t transports pupils to a bus stop of school s. In this case we call t a *school trip* for s (see Fig. 3). The time difference between the departure at the first bus stop of t and the arrival at the bus stop of s is denoted by $\delta_{st}^{\text{school}} \in \mathbb{Z}_+$. There is another time window for the pupils $\underline{\omega}_{st}^{\text{school}}$, $\bar{\omega}_{st}^{\text{school}} \in \mathbb{Z}_+$, $\underline{\omega}_{st}^{\text{school}} \leqslant \bar{\omega}_{st}^{\text{school}}$, specifying the minimal and maximal waiting time relative to the school starting time. The lower bound $\underline{\omega}_{st}^{\text{school}}$ is chosen according to the walking time from the bus stop where the pupils are dropped off, whereas the upper bound $\bar{\omega}_{st}^{\text{school}}$ is due to law restrictions. A typical time window is 5–45 minutes. Deviations from these values are often necessary due to the school type or the age of the pupils.

Let $\mathscr{C} \subset \mathscr{V} \times \mathscr{V}$ be the set of pairs (t_1,t_2), where t_1 is a trip that transports pupils to a so-called *changing bus stop*, where they leave the bus and transfer to trip t_2. We say, t_1 is a *feeder trip* for t_2 and, vice versa, t_2 is a *collector trip* for t_1 (see Fig. 4). The driving time from the first bus stop of feeder trip t_1 to the changing bus stop is denoted by $\delta_{t_1t_2}^{\text{feeder}} \in \mathbb{Z}_+$. The corresponding parameter for the collector trip is $\delta_{t_1t_2}^{\text{collector}} \in \mathbb{Z}_+$. A time window $\underline{\omega}_{t_1t_2}^{\text{change}}$, $\bar{\omega}_{t_1t_2}^{\text{change}} \in \mathbb{Z}_+$, $\underline{\omega}_{t_1t_2}^{\text{change}} \leqslant \bar{\omega}_{t_1t_2}^{\text{change}}$ for the minimal and maximal waiting time at the changing bus stop is given. Typically this time window is 0–10 minutes.

Note that a trip can have more than one type, e.g., it can be both a school and feeder trip. All trips that are neither school, feeder nor collector trips are called *free trips*. Free trips are obviously not important for the transport of pupils. However, they also have to be served by some bus. The time windows for these trips are usually quite narrow, for example plus/minus 15 minutes relative to the current starting time of the respective trip.

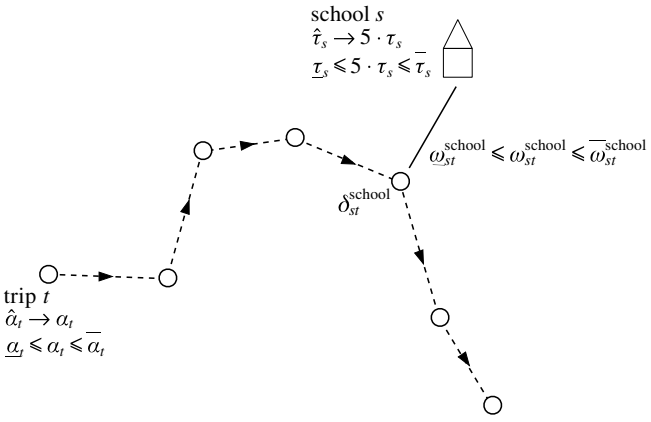

Fig. 3. School trip t for school s.

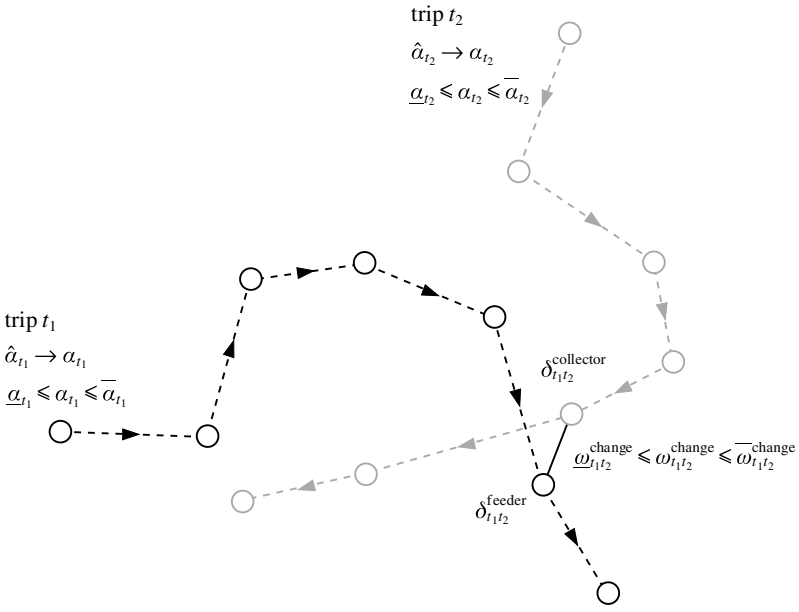

Fig. 4. A feeder trip t_1 and a collector trip t_2 with $(t_1,t_2) \in \mathscr{C}$.

2.2. Variables and bounds

Our model contains the following binary and integer variables. See Table 2 for an overview.

For every trip $t \in \mathscr{V}$ the decision variables $v_t, w_t \in \{0,1\}$ indicate if trip t is the first or the last trip in some block, respectively. For every pair of trips $(t_1,t_2) \in \mathscr{A}$ the variable $x_{t_1t_2} \in \{0,1\}$ indicates if t_1 and t_2 are in sequence in some block, that is, the same bus serves trip t_2 directly after finishing trip t_1 (apart from the deadhead trip and some idle time).

For every trip $t \in \mathscr{V}$ we introduce an integer variable $\alpha_t \in \mathbb{Z}$ representing its planned starting time, i.e., the departure of a bus at the first bus stop. Bounds on these variables are given by the corresponding time windows:

$$\underline{\alpha}_t \leqslant \alpha_t \leqslant \bar{\alpha}_t. \tag{1}$$

The school starting time is required to be in discrete time slots of 5 minutes (7:30, 7:35, 7:40, etc.). For every school $s \in \mathscr{S}$ we introduce an integer variable $\tau_s \in \mathbb{Z}$ with

$$\underline{\tau}_s \leqslant 5 \cdot \tau_s \leqslant \bar{\tau}_s. \tag{2}$$

Thus, the planned school starting time of s is given by $5 \cdot \tau_s$.

An assignment of values to the variables τ, α, v, w, x is called a *planned solution*.

2.3. Constraints

Each trip is served by exactly one bus. That means, trip $t_2 \in \mathscr{V}$ either has a unique predecessor or it is the first one in some block:

$$\sum_{t_1:(t_1,t_2)\in\mathscr{A}} x_{t_1t_2} + v_{t_2} = 1. \tag{3}$$

Moreover, every trip $t_1 \in \mathscr{V}$ either has a unique successor or it is the last one in some block:

$$\sum_{t_2:(t_1,t_2)\in\mathscr{A}} x_{t_1t_2} + w_{t_1} = 1. \tag{4}$$

If trips $(t_1,t_2) \in \mathscr{A}$ are connected, then trip t_2 can only start after the bus finished trip t_1, shifted from the end of t_1 to the start of t_2, and waited a specified time to absorb possible delays. Additional waiting is permitted within certain limits if the bus arrives before the start of t_2. We can formulate this constraint in non-linear terms as follows:

$$\begin{aligned} &\left(\alpha_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2}\right) \cdot x_{t_1t_2} \leqslant \alpha_{t_2}, \\ &\alpha_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \bar{\omega}^{\text{idle}}_{t_1t_2} \geqslant \alpha_{t_2} \cdot x_{t_1t_2}. \end{aligned} \tag{5}$$

Now using a sufficiently big value for M, these constraints can be formulated as linear inequalities:

$$\begin{aligned} &\alpha_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2} - M \cdot (1 - x_{t_1t_2}) \leqslant \alpha_{t_2}, \\ &\alpha_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \bar{\omega}^{\text{idle}}_{t_1t_2} + M \cdot (1 - x_{t_1t_2}) \geqslant \alpha_{t_2}. \end{aligned} \tag{6}$$

For the time being it is enough to think of M as a large but finite number. In fact, M actually depends on $(t_1,t_2) \in \mathscr{A}$, so every inequality has its own $M_{t_1t_2}$. Later in Section 3.3, we compute best possible values for $M_{t_1t_2}$.

For every pair $(t_1,t_2) \in \mathscr{C}$ the starting times of both bus trips must be synchronized in such a way that t_2 arrives at the changing bus stop after trip t_1 within a small time window. This is assured by the following inequalities:

$$\begin{aligned} &\alpha_{t_1} + \delta^{\text{feeder}}_{t_1t_2} + \underline{\omega}^{\text{change}}_{t_1t_2} \leqslant \alpha_{t_2} + \delta^{\text{collector}}_{t_1t_2}, \\ &\alpha_{t_1} + \delta^{\text{feeder}}_{t_1t_2} + \bar{\omega}^{\text{change}}_{t_1t_2} \geqslant \alpha_{t_2} + \delta^{\text{collector}}_{t_1t_2}. \end{aligned} \tag{7}$$

For every $(s,t) \in \mathscr{P}$ the starting times of trip t and school s have to be chosen such that the waiting time restrictions for the pupils at school s are met. Thus, we add the following inequalities to the model in order to synchronize the start of bus trips and schools:

Table 2
Variables

$v_t \in \{0,1\}$	First trip in block
$w_t \in \{0,1\}$	Last trip in block
$x_{t_1t_2} \in \{0,1\}$	Connected trips in block
$\tau_s \in \mathbb{Z} \cap [\frac{1}{5}\underline{\tau}_s, \frac{1}{5}\bar{\tau}_s]$	School starting time (time slot)
$\alpha_t \in \mathbb{Z} \cap [\underline{\alpha}_t, \bar{\alpha}_t]$	Trip starting time

$$\begin{aligned} \alpha_t + \delta^{\text{school}}_{st} + \underline{\omega}^{\text{school}}_{st} &\leqslant 5 \cdot \tau_s, \\ \alpha_t + \delta^{\text{school}}_{st} + \bar{\omega}^{\text{school}}_{st} &\geqslant 5 \cdot \tau_s. \end{aligned} \tag{8}$$

2.4. Objective function

The optimization basically has two goals: Finding a minimum number of buses to serve all trips, and deploying these buses in the most efficient way, that is, minimize the sum of all deadhead trips. Thus as objective functions we have:

- The total number of deployed vehicles:

$$\sum_{t \in \mathscr{V}} v_t, \tag{9}$$

- and the driving times of all pull-out, pull-in, and deadhead trips:

$$\sum_{t \in \mathscr{V}} \delta^{\text{out}}_t \cdot v_t + \sum_{(t_1,t_2) \in \mathscr{A}} \delta^{\text{shift}}_{t_1 t_2} \cdot x_{t_1 t_2} + \sum_{t \in \mathscr{V}} \delta^{\text{in}}_t \cdot w_t. \tag{10}$$

We turn these two objective functions (9) and (10) into a single one. To this end, we add both objectives (9) and (10), and multiply (10) by a some small number $\varepsilon > 0$, which reflects the fact that saving buses is much more important than saving deadhead trips:

$$\sum_{t \in \mathscr{V}} v_t + \varepsilon \left(\sum_{t \in \mathscr{V}} \delta^{\text{out}}_t \cdot v_t + \sum_{(t_1,t_2) \in \mathscr{A}} \delta^{\text{shift}}_{t_1 t_2} \cdot x_{t_1 t_2} + \sum_{t \in \mathscr{V}} \delta^{\text{in}}_t \cdot w_t \right). \tag{11}$$

In theory, we can set

$$\varepsilon := \sum_{t_1 \in \mathscr{V}} \Big(\max_{t_0 : (t_0,t_1) \in \mathscr{A}} \delta^{\text{shift}}_{t_0 t_1} + \max_{t_2 : (t_1,t_2) \in \mathscr{A}} \delta^{\text{shift}}_{t_1 t_2} \Big), \tag{12}$$

which gives a sufficiently small value for ε. This formula says that it is still cheaper to use one bus to serve all trips even on the longest possible deadhead trips, instead of two buses. However, this value of ε might be so small that the solver software runs into numerical problems. Thus in practice, we set $\varepsilon := \frac{1}{10000}$, meaning that deadhead trips are no longer than 120 minutes and no bus will serve more than 20 trips in a block. For the real-world instances we discuss later on these assumptions turned out to be reasonable.

3. Improving the LP relaxation of the model

Summing up we have the following integer optimization problem:

$$\begin{aligned} &\min && (11), \\ &\text{subject to} && (1)\text{–}(8), \\ & && v, w \in \{0,1\}^{|\mathscr{V}|},\ x \in \{0,1\}^{|\mathscr{A}|}, \\ & && \tau \in \mathbb{Z}^{|\mathscr{S}|},\ \alpha \in \mathbb{Z}^{|\mathscr{V}|}. \end{aligned} \tag{13}$$

The above model is a generalization of VRPTW. Since VRPTW is NP-hard we cannot expect a polynomial algorithm for its solution unless $P = \text{NP}$. Thus, we use branch-and-cut to find good feasible solutions with proven lower bound, or even optimal solutions. The most common way to obtain a lower bound is to use the LP relaxation, where the integrality constraints of (13) are relaxed (see [11] for a survey). However, for large real-world instances or instances with large time windows (see Section 5) this bound is sometimes of poor quality. In the remainder of this section, we present several ways to improve the lower bound.

3.1. Starting time propagation

School and trip starting times are coupled by the minimum and maximum waiting time restrictions of inequalities (7) or (8). The time window for the starting time of school s can be propagated onto the starting time window of trip t for all $(s,t) \in \mathscr{P}$. From (8) and the trivial constraints on the bounds of the variables (2) we obtain:

$$\alpha_t \leqslant 5 \cdot \tau_s - \delta^{\text{school}}_{st} - \underline{\omega}^{\text{school}}_{st} \leqslant \bar{\tau}_s - \delta^{\text{school}}_{st} - \underline{\omega}^{\text{school}}_{st}. \tag{14}$$

We now compare the right-hand side of (14) with the previous upper bound $\bar{\alpha}_t$ on α_t. If it is less, a new *improved* upper bound is found. In general, we set for all $(s,t) \in \mathscr{P}$:

$$\bar{\alpha}_t := \min\{\bar{\alpha}_t, \bar{\tau}_s - \delta^{\text{school}}_{st} - \underline{\omega}^{\text{school}}_{st}\}. \tag{15}$$

In the same way an improved lower bound can be derived for all $(s,t) \in \mathscr{P}$:

$$\underline{\alpha}_t := \max\{\underline{\alpha}_t, \underline{\tau}_s - \delta^{\text{school}}_{st} - \bar{\omega}^{\text{school}}_{st}\}. \tag{16}$$

Vice versa, the trip time window can be propagated onto the school time window by (1) and (8). Since the school starting times are required to be in discrete 5-min time slots, rounding has an additional influence on the time windows. With $\lceil a \rceil_b := \left\lceil \frac{a}{b} \right\rceil \cdot b$, $\lfloor a \rfloor_b := \left\lfloor \frac{a}{b} \right\rfloor \cdot b$ for $a, b \in \mathbb{Q}, b \neq 0$ we obtain for all $(s,t) \in \mathscr{P}$:

$$\begin{aligned} \underline{\tau}_s &:= \max\left\{\underline{\tau}_s, \left\lceil \underline{\alpha}_t + \delta^{\text{school}}_{st} + \underline{\omega}^{\text{school}}_{st} \right\rceil_5\right\}, \\ \bar{\tau}_s &:= \min\left\{\bar{\tau}_s, \left\lfloor \bar{\alpha}_t + \delta^{\text{school}}_{st} + \bar{\omega}^{\text{school}}_{st} \right\rfloor_5\right\}. \end{aligned} \tag{17}$$

The same idea can be applied for the time windows of feeder and collector trips, which are coupled by (7). Then the improved bounds for all $(t_1, t_2) \in \mathscr{C}$ are given by

$$\begin{aligned} \bar{\alpha}_{t_1} &:= \min\left\{\bar{\alpha}_{t_1}, \bar{\alpha}_{t_2} + \delta^{\text{collector}}_{t_1t_2} - \delta^{\text{feeder}}_{t_1t_2} - \underline{\omega}^{\text{change}}_{t_1t_2}\right\}, \\ \underline{\alpha}_{t_2} &:= \max\left\{\underline{\alpha}_{t_2}, \underline{\alpha}_{t_1} - \delta^{\text{collector}}_{t_1t_2} + \delta^{\text{feeder}}_{t_1t_2} + \underline{\omega}^{\text{change}}_{t_1t_2}\right\}, \\ \bar{\alpha}_{t_2} &:= \min\left\{\bar{\alpha}_{t_2}, \bar{\alpha}_{t_1} - \delta^{\text{collector}}_{t_1t_2} + \delta^{\text{feeder}}_{t_1t_2} + \bar{\omega}^{\text{change}}_{t_1t_2}\right\}, \\ \underline{\alpha}_{t_1} &:= \max\left\{\underline{\alpha}_{t_1}, \underline{\alpha}_{t_2} + \delta^{\text{collector}}_{t_1t_2} - \delta^{\text{feeder}}_{t_1t_2} - \bar{\omega}^{\text{change}}_{t_1t_2}\right\}. \end{aligned} \tag{18}$$

Bounds strengthening steps (15)–(18) are iteratively repeated, until either no bound can be improved any more or an infeasibility occurs. That is, there is either a trip $t \in \mathscr{V}$ with $\underline{\alpha}_t > \bar{\alpha}_t$ or a school s with $\underline{\tau}_s > \bar{\tau}_s$ after applying one of the above update steps. An infeasibility means that there is no solution for the given instance due to a conflict of some time windows. If the algorithm terminates without an infeasibility we have strengthened bounds on the α and τ variables.

The iterated application of formulas (15)–(18) is called *starting time propagation*. The starting time propagation is performed in the root LP and also in every node of the branch-and-bound tree. We remark that starting time propagation is a special case of the more general bounds strengthening preprocessing routine which is standard in most state of the art numerical solvers for mixed-integer programming problems (see [11] for further details). However, we needed to re-implement this routine in order to make further non-standard preprocessing possible (such as lifting and big-M-free reformulations described in Sections 3.5 and 3.6, respectively).

3.2. Variable fixing

After the starting time propagation, it is often possible to fix some decision variables to their respective bounds. Consider some deadhead trip $(t_1, t_2) \in \mathscr{A}$. If the upper bound $\bar{\alpha}_{t_2}$ is tightened such that trip t_2 cannot be connected with trip t_1, even if t_1 starts as early as possible, then by inequality (6) the corresponding deadhead trip can be removed from $\mathscr{A}$, and the corresponding decision variable is fixed to zero. That is, if $\underline{\alpha}_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2} > \bar{\alpha}_{t_2}$, then $x_{t_1t_2} = 0$. Consider a trip $t_2 \in \mathscr{V}$. If we can fix $x_{t_1t_2} = 0$ for all preceding trips $t_1 \in \mathscr{V}$ with $(t_1, t_2) \in \mathscr{A}$, then the upper bound on the starting time of trip t_2 was too low for a bus to serve trip t_2 after having already served any other trip. Hence, a new bus from the depot is required to serve trip t_2, i.e., t_2 is the first trip in some block, and v_{t_2} can be fixed to 1 by (3). Vice versa, consider a trip $t_1 \in \mathscr{V}$ where we fixed $x_{t_1t_2} = 0$ for all succeeding trips t_2 with $(t_1, t_2) \in \mathscr{A}$, then the lower bound $\underline{\alpha}_{t_1}$ on the starting time of trip t_1 was too high so that a bus cannot serve any other trip after serving this trip. Hence, the bus is sent back to the depot, i.e., trip t_1 is the last trip in some block. Thus, w_{t_1} can be fixed to 1 by (4).

3.3. Big-M reduction

Consider the inequalities in (6). Informally speaking, due to the "big-M"-formulation they have the disadvantage of not being very strong, that is, even when they are removed from the model, the objective function value of the LP relaxation does not change much. The x and the α variables are only loosely linked in the LP relaxation. Moreover, it is known that models having "big-M"-inequalities tend to cause computational problems. One way to reduce (but not

eliminate) the problem is to compute best possible values for M so that the corresponding inequalities become as strong as possible.

Let $(t_1, t_2) \in \mathscr{A}$. It was already noted that M depends on (t_1, t_2). We strengthen these inequalities by computing the best possible value for $M_{t_1t_2}$. The first inequality in (6) is weak for $x_{t_1t_2} = 0$ in the sense that for all $\alpha_{t_1} \in [\underline{\alpha}_{t_1}, \bar{\alpha}_{t_1}]$ and all $\alpha_{t_2} \in [\underline{\alpha}_{t_2}, \bar{\alpha}_{t_2}]$, we have:

$$\alpha_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2} - M'_{t_1t_2} \leqslant \alpha_{t_2}. \tag{19}$$

Therefore, we get:

$$\bar{\alpha}_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2} - M'_{t_1t_2} \leqslant \underline{\alpha}_{t_2}, \tag{20}$$

thus we can set:

$$M'_{t_1t_2} := \bar{\alpha}_{t_1} - \underline{\alpha}_{t_2} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2}. \tag{21}$$

For the second inequality in (6), we obtain by similar computations:

$$M''_{t_1t_2} := \bar{\alpha}_{t_2} - \underline{\alpha}_{t_1} - \delta^{\text{trip}}_{t_1} - \delta^{\text{shift}}_{t_1t_2} - \bar{\omega}^{\text{idle}}_{t_1t_2}. \tag{22}$$

Note that these "big-M"-computations are carried out *after* the starting times propagation, because there the bounds on α might be changed. This affects (improves) also the values for $M'_{t_1t_2}$ and $M''_{t_1t_2}$, respectively.

3.4. Higher order variable fixing

In the above variable fixing routine, only one inequality is used after the other to fix binary variables to their lower bound. Much stronger results can be obtained if more than one inequality is taken into consideration (two or three, for example). This is what we call *higher order variable fixing*.

Consider two trips $(t_1, t_2) \in \mathscr{A}$ with $(t_1, t_2) \in \mathscr{C}$. Here the relevant subsystem of (13) consists of the first inequality of (6) and the second inequality of (7). That is,

$$\begin{aligned} &\alpha_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2} - M \cdot (1 - x_{t_1t_2}) \leqslant \alpha_{t_2}, \\ &\alpha_{t_1} + \delta^{\text{feeder}}_{t_1t_2} + \bar{\omega}^{\text{change}}_{t_1t_2} \geqslant \alpha_{t_2} + \delta^{\text{collector}}_{t_1t_2}. \end{aligned} \tag{23}$$

By adding up these two inequalities we can eliminate the variables α_{t_1}, α_{t_2} and obtain

$$\delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2} - \delta^{\text{feeder}}_{t_1t_2} - \bar{\omega}^{\text{change}}_{t_1t_2} + \delta^{\text{collector}}_{t_1t_2} \leqslant M \cdot (1 - x_{t_1t_2}). \tag{24}$$

Hence if $\delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2} - \delta^{\text{feeder}}_{t_1t_2} - \bar{\omega}^{\text{change}}_{t_1t_2} + \delta_{t_1t_2}{}^{\text{collector}} > 0$ we set $x_{t_1t_2} := 0$ and thus eliminate this decision variable.

Using the second inequality of (6) and the first inequality of (7) we obtain after similar computations that we can set $x_{t_1t_2} := 0$ if $\delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \bar{\omega}^{\text{idle}}_{t_1t_2} - \delta^{\text{feeder}}_{t_1t_2} - \underline{\omega}^{\text{change}}_{t_1t_2} + \delta^{\text{collector}}_{t_1t_2} < 0$.

Consider two trips $(t_1, t_2) \in \mathscr{A}$ with $(s, t_1), (s, t_2) \in \mathscr{P}$ for some school $s \in \mathscr{S}$. The relevant subsystem of (13) consists of the first inequality of (6), the second inequality of (8) for (s, t_1), and the first inequality of (8) for (s, t_2). That is,

$$\begin{aligned} &\alpha_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2} - M \cdot (1 - x_{t_1t_2}) \leqslant \alpha_{t_2}, \\ &\alpha_{t_1} + \delta^{\text{school}}_{st_1} + \bar{\omega}^{\text{school}}_{st_1} \geqslant 5 \cdot \tau_s, \\ &\alpha_{t_2} + \delta^{\text{school}}_{st_2} + \underline{\omega}^{\text{school}}_{st_2} \leqslant 5 \cdot \tau_s. \end{aligned} \tag{25}$$

By adding up these three inequalities we can eliminate the variables α_{t_1}, α_{t_2}, τ_s and obtain:

$$\delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2} - \delta^{\text{school}}_{st_1} - \bar{\omega}^{\text{school}}_{st_1} + \delta^{\text{school}}_{st_2} + \underline{\omega}^{\text{school}}_{st_2} \leqslant M \cdot (1 - x_{t_1t_2}). \tag{26}$$

Thus, if $\delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2} - \delta^{\text{school}}_{st_1} - \bar{\omega}^{\text{school}}_{st_1} + \delta^{\text{school}}_{st_2} + \underline{\omega}^{\text{school}}_{st_2} > 0$ we set $x_{t_1t_2} := 0$.

Using the second inequality of (6), the first inequality of (8) for (s, t_1), and the second inequality of (8) for (s, t_2) we obtain that we can set $x_{t_1t_2} := 0$ if:

$$\delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \bar{\omega}^{\text{idle}}_{t_1t_2} - \delta^{\text{school}}_{st_1} - \underline{\omega}^{\text{school}}_{st_1} + \delta^{\text{school}}_{st_2} + \bar{\omega}^{\text{school}}_{st_2} < 0. \tag{27}$$

3.5. Lifting coefficients

Inequalities (6) can be further strengthened. For this we apply a lifting technique developed by Desrochers and Laporte [9] (see also [13]) for the vehicle routing problem with time windows, VRPTW. (More general, lifting as a technique for

general integer programming to strengthen valid inequalities was introduced by Padberg [18] and Wolsey [22].) The strengthening is based on the observation that $x_{t_1t_2} + x_{t_2t_1} \leqslant 1$ for all $(t_1, t_2) \in \mathscr{A}$ with also $(t_2, t_1) \in \mathscr{A}$.

Theorem 1. *For all* $(t_1, t_2) \in \mathscr{A}$ *with* $(t_2, t_1) \in \mathscr{A}$ *the constraints*:

$$\alpha_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2} - \left(\bar{\alpha}_{t_1} - \underline{\alpha}_{t_2} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2}\right) \cdot (1 - x_{t_1t_2}) + \max\left\{0, \bar{\alpha}_{t_1} - \underline{\alpha}_{t_2} - \delta^{\text{trip}}_{t_2} - \delta^{\text{shift}}_{t_2t_1} - \bar{\omega}^{\text{idle}}_{t_2t_1}\right\} \cdot x_{t_2t_1} \leqslant \alpha_{t_2} \tag{28}$$

and

$$\alpha_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \bar{\omega}^{\text{idle}}_{t_1t_2} + \left(\bar{\alpha}_{t_2} - \underline{\alpha}_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \bar{\omega}^{\text{idle}}_{t_1t_2}\right) \cdot (1 - x_{t_1t_2}) - \max\left\{0, \bar{\alpha}_{t_2} - \underline{\alpha}_{t_1} - \delta^{\text{trip}}_{t_2} - \delta^{\text{shift}}_{t_2t_1} - \underline{\omega}^{\text{idle}}_{t_2t_1}\right\} \cdot x_{t_2t_1} \geqslant \alpha_{t_2} \tag{29}$$

are valid inequalities for (13).

Proof. Consider for an arbitrary $(t_1, t_2) \in \mathscr{A}$ with $(t_2, t_1) \in \mathscr{A}$ inequality (6) with $M'_{t_1t_2}$ as "big-M", that is,

$$\alpha_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2} - \left(\bar{\alpha}_{t_1} - \underline{\alpha}_{t_2} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2}\right) \cdot (1 - x_{t_1t_2}) \leqslant \alpha_{t_2}. \tag{30}$$

We introduce an additional term $\xi_{t_2t_1} \cdot x_{t_2t_1}$ in (30) and obtain:

$$\alpha_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2} - \left(\bar{\alpha}_{t_1} - \underline{\alpha}_{t_2} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2}\right) \cdot (1 - x_{t_1t_2}) + \xi_{t_2t_1} \cdot x_{t_2t_1} \leqslant \alpha_{t_2}. \tag{31}$$

We are seeking the "best" (i.e., largest) value for $\xi_{t_2t_1}$. The new term only occurs when $x_{t_2t_1} = 1$. In this case we have $x_{t_1t_2} = 0$. Thus (31) reads:

$$\alpha_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2} - \left(\bar{\alpha}_{t_1} - \underline{\alpha}_{t_2} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2}\right) + \xi_{t_2t_1} \leqslant \alpha_{t_2}, \tag{32}$$

hence $\xi_{t_2t_1}$ must be chosen in such way that:

$$\xi_{t_2t_1} \leqslant \alpha_{t_2} - \alpha_{t_1} + \bar{\alpha}_{t_1} - \underline{\alpha}_{t_2} \tag{33}$$

is fulfilled. From inequality (6) for (t_2, t_1), we obtain with $x_{t_2t_1} = 1$:

$$\alpha_{t_2} + \delta^{\text{trip}}_{t_2} + \delta^{\text{shift}}_{t_2t_1} + \bar{\omega}^{\text{idle}}_{t_2t_1} + M \cdot (1 - 1) \geqslant \alpha_{t_1}, \tag{34}$$

and thus,

$$\alpha_{t_2} - \alpha_{t_1} \geqslant -\delta^{\text{trip}}_{t_2} - \delta^{\text{shift}}_{t_2t_1} - \bar{\omega}^{\text{idle}}_{t_2t_1}. \tag{35}$$

So we can estimate:

$$\bar{\alpha}_{t_1} - \underline{\alpha}_{t_2} - \delta^{\text{trip}}_{t_2} - \delta^{\text{shift}}_{t_2t_1} - \bar{\omega}^{\text{idle}}_{t_2t_1} \leqslant \alpha_{t_2} - \alpha_{t_1} + \bar{\alpha}_{t_1} - \underline{\alpha}_{t_2}. \tag{36}$$

Now we set:

$$\xi_{t_2t_1} := \bar{\alpha}_{t_1} - \underline{\alpha}_{t_2} - \delta^{\text{trip}}_{t_2} - \delta^{\text{shift}}_{t_2t_1} - \bar{\omega}^{\text{idle}}_{t_2t_1}, \tag{37}$$

which yields the proposition. (Note that for $\xi_{t_2t_1} \leqslant 0$, the coefficient does not need to be lifted into the inequality.)

The proof for inequality (29) is similar, thus we skip the details. □

Replacing (6) by (28) and (29), respectively, leads to a tighter LP relaxation of the model.

The next theorem is also inspired by the article of Desrochers and Laporte [9]. Here we take the trivial inequalities $\underline{\alpha} \leqslant \alpha \leqslant \bar{\alpha}$ and introduce x-variables by lifting.

Theorem 2. *For all* $t_1 \in \mathscr{V}$ *the constraints*:

$$\underline{\alpha}_{t_1} + \sum_{t_0:(t_0,t_1)\in\mathscr{A}} \max\left\{0, \underline{\alpha}_{t_0} - \underline{\alpha}_{t_1} + \delta^{\text{trip}}_{t_0} + \delta^{\text{shift}}_{t_0t_1} + \underline{\omega}^{\text{idle}}_{t_0t_1}\right\} \cdot x_{t_0t_1} \leqslant \alpha_{t_1}, \tag{38}$$

$$\bar{\alpha}_{t_1} - \sum_{t_0:(t_0,t_1)\in\mathscr{A}} \max\left\{0, \underline{\alpha}_{t_1} - \underline{\alpha}_{t_0} - \delta^{\text{trip}}_{t_0} - \delta^{\text{shift}}_{t_0t_1} - \bar{\omega}^{\text{idle}}_{t_0t_1}\right\} \cdot x_{t_0t_1} \geqslant \alpha_{t_1}, \tag{39}$$

$$\bar{\alpha}_{t_1} - \sum_{t_2:(t_1,t_2)\in\mathscr{A}} \max\left\{0, \bar{\alpha}_{t_1} - \bar{\alpha}_{t_2} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2}\right\} \cdot x_{t_1t_2} \geqslant \alpha_{t_1}, \tag{40}$$

$$\underline{\alpha}_{t_1} + \sum_{t_2:(t_1,t_2)\in\mathscr{A}} \max\left\{0, \underline{\alpha}_{t_2} - \underline{\alpha}_{t_1} - \delta^{\text{trip}}_{t_1} - \delta^{\text{shift}}_{t_1t_2} - \bar{\omega}^{\text{idle}}_{t_1t_2}\right\} \cdot x_{t_1t_2} \leqslant \alpha_{t_1} \tag{41}$$

are valid inequalities for (13).

Proof. To prove inequality (38), let $t_2 \in \mathscr{V}$. By the lower bound inequality of (1) on the trip starting time we have $\underline{\alpha}_{t_2} \leqslant \alpha_{t_2}$. For each $t_1 \in \mathscr{V}$ with $(t_1, t_2) \in \mathscr{A}$, we introduce an additional term $\xi_{t_1t_2} \cdot x_{t_1t_2}$ and compute the largest possible value for $\xi_{t_1t_2}$. Note that these computations can be done sequentially, because for at most one t_1 we have $x_{t_1t_2} = 1$, thus all $\xi_{t_1t_2}$ are independent of each other. For $x_{t_1t_2} = 0$, we can assume an arbitrary value for $\xi_{t_1t_2}$. For $x_{t_1t_2} = 1$, we obtain from the first inequality of (6):

$$\alpha_{t_2} \geqslant \alpha_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2} \geqslant \underline{\alpha}_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2} = \underline{\alpha}_{t_2} + \left(-\underline{\alpha}_{t_2} + \underline{\alpha}_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2}\right). \tag{42}$$

If $-\underline{\alpha}_{t_2} + \underline{\alpha}_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2}$ is negative then 0 is a better bound because of (1). Thus, we obtain $\xi_{t_1t_2} := \max\left\{0, -\underline{\alpha}_{t_2} + \underline{\alpha}_{t_1} + \delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2}\right\}$.

The proof of the other inequalities (39)–(41) can be carried out analogously, so we omit the details. □

We use the techniques from the previous theorem to lift coefficients in the trivial inequalities for the τ variables.

Theorem 3. *For all $(s, t_1) \in \mathscr{P}$ the constraints*:

$$\frac{1}{5}\bar{\tau}_s \cdot w_{t_1} + \sum_{t_2:(t_1,t_2)\in\mathscr{A}} \left\lfloor \bar{\alpha}_{t_2} - \delta^{\text{trip}}_{t_1} - \delta^{\text{shift}}_{t_1t_2} - \underline{\omega}^{\text{idle}}_{t_1t_2} + \delta^{\text{school}}_{st_1} + \bar{\omega}^{\text{school}}_{st_1} \right\rfloor_5 \cdot x_{t_1t_2} \geqslant \tau_s, \tag{43}$$

$$\frac{1}{5}\bar{\tau}_s \cdot v_{t_1} + \sum_{t_0:(t_0,t_1)\in\mathscr{A}} \left\lfloor \bar{\alpha}_{t_0} + \delta^{\text{trip}}_{t_0} + \delta^{\text{shift}}_{t_0t_1} + \bar{\omega}^{\text{idle}}_{t_0t_1} + \delta^{\text{school}}_{st_1} + \bar{\omega}^{\text{school}}_{st_1} \right\rfloor_5 \cdot x_{t_0t_1} \geqslant \tau_s, \tag{44}$$

$$\frac{1}{5}\underline{\tau}_s \cdot w_{t_1} + \sum_{t_2:(t_1,t_2)\in\mathscr{A}} \left\lceil \underline{\alpha}_{t_2} - \delta^{\text{trip}}_{t_1} - \delta^{\text{shift}}_{t_1t_2} - \bar{\omega}^{\text{idle}}_{t_1t_2} + \delta^{\text{school}}_{st_1} + \underline{\omega}^{\text{school}}_{st_1} \right\rceil_5 \cdot x_{t_1t_2} \leqslant \tau_s, \tag{45}$$

$$\frac{1}{5}\underline{\tau}_s \cdot v_{t_1} + \sum_{t_0:(t_0,t_1)\in\mathscr{A}} \left\lceil \underline{\alpha}_{t_0} + \delta^{\text{trip}}_{t_0} + \delta^{\text{shift}}_{t_0t_1} + \underline{\omega}^{\text{idle}}_{t_0t_1} + \delta^{\text{school}}_{st_1} + \underline{\omega}^{\text{school}}_{st_1} \right\rceil_5 \cdot x_{t_0t_1} \leqslant \tau_s \tag{46}$$

are valid inequalities for (13).

Proof. We only prove (43) since the other three inequalities can be shown similarly. Let $s \in \mathscr{S}$ be an arbitrary school and $t_1 \in \mathscr{V}$ a trip with $(s, t_1) \in \mathscr{P}$. Because of (4) there is at most one trip $t_2 \in \mathscr{V}$ with $(t_1, t_2) \in \mathscr{A}$ and $x_{t_1t_2} = 1$. Using inequalities (6) and (8), we can estimate the upper bounds on the school starting time in this case as follows:

$$5\tau_s \leqslant \alpha_{t_1} + \delta^{\text{school}}_{st_1} + \bar{\omega}^{\text{school}}_{st_1} \leqslant \bar{\alpha}_{t_2} - \delta^{\text{trip}}_{t_1} - \delta^{\text{shift}}_{t_1t_2} - \underline{\omega}^{\text{idle}}_{t_1t_2} + \delta^{\text{school}}_{st_1} + \bar{\omega}^{\text{school}}_{st_1}. \tag{47}$$

After dividing this inequality by 5 we can round down the right-hand side. That is,

$$\tau_s \leqslant \left\lfloor \bar{\alpha}_{t_2} - \delta^{\text{trip}}_{t_1} - \delta^{\text{shift}}_{t_1t_2} - \underline{\omega}^{\text{idle}}_{t_1t_2} + \delta^{\text{school}}_{st_1} + \bar{\omega}^{\text{school}}_{st_1} \right\rfloor_5. \tag{48}$$

If there is no such trip t_2 then $w_{t_1} = 1$ because of (4). In this case we have $\tau_s \leqslant \frac{1}{5}\bar{\tau}_s$. Putting this together, we obtain inequality (43). □

3.6. Big-M-free reformulation

In [1,2], Ascheuer, Fischetti, and Grötschel (who give the credit to Maffioli and Sciomachen [17], respectively, van Eijl [20]) describe a model that avoids the need of "big-M" terms (such as (6) in our model). Their reformulation was introduced to solve instances of a single-vehicle traveling salesman problem with time windows. We will now demonstrate how to apply their approach to a reformulation of our model (13).

We introduce additional integer variables $\xi_{t_1t_2} \in \mathbb{Z}_+$ for all $(t_1, t_2) \in \mathscr{A}$ and $\zeta_t \in \mathbb{Z}_+$ for all $t \in \mathscr{V}$. These new variables represent the starting time of a trip in the following way: If trip t_1 is connected with trip t_2 (that is, $x_{t_1t_2} = 1$) then $\xi_{t_1t_2}$ denotes the time when trip t_1 is starting. Otherwise if $x_{t_1t_2} = 0$ then $\xi_{t_1t_2} = 0$. If trip t is the last trip in some block (that is, $w_t = 1$) then ζ_t denotes the starting time of trip t. Otherwise if $w_t = 0$ then $\zeta_t = 0$. This is modelled by the following inequalities.

The lower and upper bounds on the starting times for the ζ, ξ variables are coupled to the decision variables x and w. For all $(t_1, t_2) \in \mathscr{A}$, we have:

$$\underline{\alpha}_{t_1} \cdot x_{t_1t_2} \leqslant \xi_{t_1t_2} \leqslant \bar{\alpha}_{t_1} \cdot x_{t_1t_2}, \tag{49}$$

and for all $t \in \mathscr{V}$, we have:

$$\underline{\alpha}_t \cdot w_t \leqslant \zeta_t \leqslant \bar{\alpha}_t \cdot w_t. \tag{50}$$

In terms of the α variables the starting time of trip $t_1 \in \mathscr{V}$ is given by

$$\alpha_{t_1} = \zeta_{t_1} + \sum_{t_2:(t_1,t_2)\in\mathscr{A}} \xi_{t_1t_2}. \tag{51}$$

Most important, inequalities (6) can be replaced by the following formulation without the use of "big-M"-terms for all $t_2 \in \mathscr{V}$:

$$\sum_{t_1:(t_1,t_2)\in\mathscr{A}} (\xi_{t_1t_2} + \left(\delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2}\right) \cdot x_{t_1t_2}) \leqslant \alpha_{t_2}. \tag{52}$$

Inequalitiy (52) can further be strengthened by introducing the lower bound on α_{t_2} as follows:

$$\underline{\alpha}_{t_2} + \sum_{t_1:(t_1,t_2)\in\mathscr{A}} \left(\xi_{t_1t_2} + \left(\delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \underline{\omega}^{\text{idle}}_{t_1t_2} - \underline{\alpha}_{t_2}\right) \cdot x_{t_1t_2}\right) \leqslant \alpha_{t_2}. \tag{53}$$

Similarly for the upper idle waiting time limit we obtain:

$$\bar{\alpha}_{t_2} + \sum_{t_1:(t_1,t_2)\in\mathscr{A}} \left(\xi_{t_1t_2} + \left(\delta^{\text{trip}}_{t_1} + \delta^{\text{shift}}_{t_1t_2} + \bar{\omega}^{\text{idle}}_{t_1t_2} - \bar{\alpha}_{t_2}\right) \cdot x_{t_1t_2}\right) \geqslant \alpha_{t_2}. \tag{54}$$

That is, if trip t_1 is connected with trip t_2 by a deadhead trip, then the starting time α_{t_2} of trip t_2 is bounded by the starting time $\xi_{t_1t_2}$ of t_1 plus the driving time for the trip t_1 and the deadhead trip (t_1, t_2) and the minimum and maximum idle time, respectively. If t_1 is not connected with t_2, then the left-hand side of inequalities (53) and (54) is equal to $\underline{\alpha}_{t_2}$ or $\bar{\alpha}_{t_2}$, respectively.

Using this big-M-free formulation, the model (13) now takes the following form:

$$\begin{aligned} &\min && (11),\\ &\text{subject to} && (1)\text{–}(4),\ (7),\ (8),\ (49)\text{–}(51),\ (53),\ (54),\\ &&& v, w \in \{0,1\}^{|\mathscr{V}|},\ x \in \{0,1\}^{|\mathscr{A}|},\\ &&& \tau \in \mathbb{Z}^{|\mathscr{S}|},\ \alpha, \zeta \in \mathbb{Z}^{|\mathscr{V}|},\ \xi \in \mathbb{Z}^{|\mathscr{A}|}. \end{aligned} \tag{55}$$

3.7. Cutting planes

In order to solve instances of the VRPTW Kohl et al. [15] (see also the survey of Cordeau et al. [7]) introduced the following cuts which we adapt to solve instances of our model (13) or (55) (the inequalities are equally valid for both models).

For a subset of the trips $\mathscr{W} \subseteq \mathscr{V}$ we define the following variable representing the flow into $\mathscr{W}$:

$$f(\mathscr{W}) := \sum_{t_1\in\mathscr{V}\setminus\mathscr{W}} \sum_{t_2\in\mathscr{W}} x_{t_1t_2} + \sum_{t_2\in\mathscr{W}} v_{t_2}. \tag{56}$$

Let $k(\mathscr{W})$ be the smallest number of buses necessary to serve all trips in $\mathscr{W}$. Then the constraint:

$$f(\mathscr{W}) \geqslant k(\mathscr{W}), \tag{57}$$

is called a k-path cut. At least k paths (pull-in or deadhead trips) have to enter subset $\mathscr{W}$ in a feasible solution for (55).

In order to use (57) as cutting planes to cut-off infeasible fractional LP-optimal solutions we have to address two problems: Given an LP-optimal solution of (55) how to quickly determine a suitable subset $\mathscr{W}$, and how to compute $k(\mathscr{W})$. (Note that the latter problem is NP-hard, since it involves the solution of an albeit smaller instance of (55) on the subset $\mathscr{W}$.)

In order to generate a suitable inequality (57) as a cutting plane we make use of several different heuristics. In all of them, we first construct a candidate subset $\mathscr{W}$ and then compute the number of vehicles that is necessary to serve the trips in $\mathscr{W}$. Let $v^*, w^*, x^* \in [0,1]$ be the optimal solution of the LP relaxation of model (13). The heuristic strategies base on the observation that in the LP relaxation "fractional subcycles" frequently occur. By this we mean a subset of trips $t_1, \ldots, t_k$ where $x^*_{t_1t_2}, x^*_{t_2t_3}, \ldots, x^*_{t_kt_1}$ are close 1, i.e., to their upper bound.

Heuristic 1. As a seed we identify a deadhead trip $(t_1, t_2) \in \mathscr{A}$ with $(t_1, t_2) = \text{argmax}\{x^*_{ij} : (i,j) \in \mathscr{A}\}$. We then start from t_2 and identify a trip t_3 with $(t_2, t_3) = \text{argmax}\{x^*_{t_2i} : (t_2, i) \in \mathscr{A}\}$. In this way, we continue building up a sequence of trips $t_1, t_2, t_3, \ldots$ until either a circle occurs, that is, for some i, j with $i < j$ we have $t_i = t_j$ within the sequence, or the length of the sequence reaches an upper limit (say 10 trips, for instance). In the first case, subset $\mathscr{W}$ is the circle in the sequence. In the latter case, subset $\mathscr{W}$ is the entire sequence.

Heuristic 2. This heuristic is a slight modification of the previous one. As a seed we now take a deadhead trip $(t_1, t_2) \in \mathscr{A}$ with $(t_1, t_2) = \operatorname{argmin}\{f(\{i,j\}) : (i,j) \in \mathscr{A}\}$. We start with $\mathscr{W} := \{t_1, t_2\}$ and identify a trip t_3 with $t_3 = \operatorname{argmin}\{f(\mathscr{W} \cup \{i\}) : i \in \mathscr{V}\}$. If $f(\mathscr{W} \cup \{t_3\}) < f(\mathscr{W}) + \varepsilon$ then we set $\mathscr{W} := \mathscr{W} \cup \{t_3\}$. These steps are now repeated until no more trip is found or $|\mathscr{W}|$ has reached a given limit (10 trips, for instance). Here ε is a parameter that takes control of the size of the generated set $\mathscr{W}$. For our numerical experiments presented below, setting $\varepsilon := 0.5$ gives a good trade off between computation time and improvement of the lower bound.

Heuristic 3. We solve an auxiliary integer program to find a candidate set $\mathscr{W}$. For this we introduce an artificial depot node 0 and let $\mathscr{V}_0 := \mathscr{V} \cup \{0\}, \mathscr{A}_0 := \mathscr{A} \cup (\{0\} \times \mathscr{V}) \cup (\mathscr{V} \times \{0\})$. The decision variables $w \in \{0,1\}^{\mathscr{V}_0}$ indicate whether trip t is in $\mathscr{W}$ (for $w_t = 1$) or not (for $w_t = 0$). The decision variables $y \in \{0,1\}^{\mathscr{A}_0}$ indicate the paths flowing into $\mathscr{W}$. We have $y_{t_1 t_2} = 1$ if and only if $t_1 \notin \mathscr{W}$ and $t_2 \in \mathscr{W}$. The y and z variables are coupled through:

$$y_{t_1 t_2} \leqslant 1 - z_{t_1}, \tag{58}$$

$$y_{t_1 t_2} \leqslant z_{t_2}, \tag{59}$$

$$z_{t_2} - z_{t_1} \leqslant y_{t_1 t_2} \tag{60}$$

for all $(t_1, t_2) \in \mathscr{A}_0$. The depot node 0 is only an artificial node. Hence it can never belong to $\mathscr{W}$:

$$z_0 = 0. \tag{61}$$

We restrict the maximum size of $\mathscr{W}$ by an upper bound constraint:

$$\sum_{t \in \mathscr{V}} z_t \leqslant M \tag{62}$$

(with $M := 10$, for instance). On the other hand, there is a minimum number of elements that should be in $\mathscr{W}$. Thus, we have a lower bound constraint:

$$\sum_{t \in \mathscr{V}} z_t \geqslant m. \tag{63}$$

The objective is to find a subset $\mathscr{W}$ with minimum inflow. The (heuristic) hope is that this would give a promising candidate for a k-path cut. As objective function we thus use:

$$\sum_{t \in \mathscr{V}} v^*_t y_{0t} + \sum_{(t_1,t_2) \in \mathscr{A}} x^*_{t_1 t_2} y_{t_1 t_2} + \sum_{t \in \mathscr{V}} w^*_t y_{t0}. \tag{64}$$

Summing up, we solve the following integer program:

$$\begin{aligned} F_{m,M} = \min \quad & (64), \\ \text{subject to } & (58)\text{–}(63), \\ & y \in \{0,1\}^{\mathscr{A}_0},\ z \in \{0,1\}^{\mathscr{V}_0}. \end{aligned} \tag{65}$$

We solve (65) using a standard branch-and-cut MIP-solver to optimality. Denote y^*, z^* the optimal solution and let $\mathscr{W} := \{t \in \mathscr{V} : z^*_t = 1\}$. Then by definition we have $F_{m,M} = f(\mathscr{W})$.

For the second problem, the computation of $k(\mathscr{W})$, we solve to optimality an instance of (55) restricted to the set of trips $\mathscr{W}$. Since we restrict ourselves to small sets $\mathscr{W}$ with typically no more than 10 trips, the numerical solution can in practice be obtained rather quickly (typically in less than 1 second). If we have $k(\mathscr{W}) > f(\mathscr{W})$, a violated k-path cut is found and added to strengthen the LP relaxation.

4. Input data

Data sets from five different German counties were available, where we were involved in consulting projects. To give an impression of these counties, we quote some facts from Wikipedia [21]. `Demmin` is a "Kreis" (district) in Mecklenburg-Western Pomerania. The county has 92,935 inhabitants and an area of 1921 km^2. `Steinfurt` is a district in the northern part of North Rhine-Westphalia. The county has 438,765 inhabitants and an area of 1792 km^2. `Soest` is also in nothern North Rhine-Westphalia, having 307,809 inhabitants and an area of 1327 km^2. `Wernigerode` is a district in the west of Saxony-Anhalt, with 94,556 inhabitants and an area of 798 km^2. Finally, `Guetersloh` is a county also in North Rhine-Westphalia, where 345,370 people live on an area of 967 km^2. The sizes of the corresponding problem instances are shown in Table 3. Around 10% of the inhabitants are pupils, half of them use public buses to get to school.

Table 3
Sizes of the sets and the IP for the five real-world instances

County	$\lvert\mathscr{V}\rvert$	$\lvert\mathscr{A}\rvert$	$\lvert\mathscr{S}\rvert$	$\lvert\mathscr{P}\rvert$	$\lvert\mathscr{C}\rvert$	Variables	Constraints
Demmin	247	60,762	43	195	165	61,589	62,019
Steinfurt	490	239,610	102	574	406	241,284	242,652
Soest	191	36,290	82	294	182	37,027	37,706
Wernigerode	134	17,822	37	201	204	18,298	18,937
Guetersloh	404	162,812	84	579	708	166,108	166,194

Table 4
Sizes of the randomly generated input data sets and MIP models

	$\lvert\mathscr{V}\rvert$	$\lvert\mathscr{A}\rvert$	$\lvert\mathscr{S}\rvert$	$\lvert\mathscr{P}\rvert$	$\lvert\mathscr{C}\rvert$	Variables	Constraints
rnd_1	25	600	10	36	19	685	770
rnd_2	25	600	10	35	15	690	758
rnd_3	25	600	10	37	17	685	760

As one can imagine, transforming input data into problem data is a complicated and time consuming process, mostly because it involves several steps of digitalization and data correction that can only be carried out manually. It is thus desirable to have instances with designable properties at hand: Given a number of schools, trips, bus stops, etc., one wants to create an instance with exactly these characteristics. In addition, the produced instance should not look too artificial. In contrast to the direct and somehow uncontrolled generation of random numbers as input figures, the main idea behind our generator of instances is to produce an entire county randomly, with smaller and bigger cities, streets between them, high schools in big cities, primary schools in the villages, and bus trips between cities and villages. We developed an instance generator according to these guidelines (for the details see [10]). With it, three random instances rnd_1 to rnd_3 were generated, see Table 4. They are much smaller than the real-world instances, and consist of only 25 trips and 10 schools each. The instance generator was called three times with exactly the same settings. The number of trips with pupils and the number of transfers are more or less identical. The last two columns show the sizes (i.e., number of variables and constraints) of the integer programming models (13) corresponding to the instances. Later we will see that there is a great variety in the time needed to solve these three instance, although they are nearly of the same size.

5. Computational results

All methods described in the previous section were implemented in C++ (gnu compiler) on a personal computer running Debian Linux as operational system, 2 GByte RAM, and an Intel IV 2.6 GHz CPU. As LP and MIP solver we use CPLEX 9.0 together with ILOG Concert Technology 2.0 (see [12]).

The first thing to do when dealing with a new instance (no matter if random or real-world) is to get an idea how many buses have to be deployed when the trip and school starting times are not changed. This is done by fixing all non-binary variables, i.e., α and τ, to the values of the current solution: $\alpha_t = \hat{\alpha}_t$ for all $t \in \mathscr{V}$ and $\tau_s = \hat{\tau}_s$ for all $s \in \mathscr{S}$. The remaining problem consists only of the binary variables x, v, w, and is computationally easy to solve. On our computer environment this takes less than a second. The results for the three instances are given in Table 5. For example, in instance rnd_1 there are currently 24 buses used to serve the 25 trips of the instance. Altogether we have 2 minutes for deadhead trips.

For each of the three random instances we specify three different school time windows. In rnd_1a, rnd_2a and rnd_3a, schools may start between 7:45 and 8:15, in rnd_1b, rnd_2b and rnd_3b between 7:30 and 8:30, and in rnd_1c, rnd_2c and rnd_3c between 7:15 and 8:45. Clearly, the wider the school starting time window, the potentially lower the number of deployed vehicles. Altogether we have nine test instances now.

We give a time limit of 3600 seconds for each instance. We tighten the time windows, fix binary variables and compute best-possible values for big-M as described in Sections 3.1–3.3, respectively. (Note that these three operations are already implemented within CPLEX, as part of the general preprocessing routine.) First we solve the LP relaxation of model (13)

Table 5
Current number of buses and deadhead trips

Instance	Objective
rnd_1	24.0002
rnd_2	25.0000
rnd_3	24.0002

for each instance. We remark that the quality of the LP relaxation (in terms of the objective function value) is better the smaller the starting time windows are. The LP relaxation is also the root node of the branch-and-bound tree. Thus in a branch-and-bound framework there are three benefits from narrow time windows: The number of possible solutions is lower (due to the fact that significantly fewer deadhead trips are available), the dual bound is better, which results in fewer nodes until an optimal solution is found, and finally the LP's are smaller (having less variables and constraints) which makes them faster to solve. The best lower and upper bounds found by the MIP solver are shown in Table 6. These results indicate what a modern state-of-the-art commercial solver for mixed-integer programs is by default able to do for our test instances. If lower and upper bound coincide then a globally optimal solution is found within the given time limit. This is the case for four of the nine instances, which are marked in **boldface**. The integrality gap in these cases is 0%. (We compute this gap as $\frac{u-l}{l}$, where u is the (best) upper bound and l is the (best) lower bound returned by the MIP solver when the time limit is reached.)

There are some binary variables that CPLEX's standard preprocessing is not able to remove from the model, since it would involve preprocessing with more than one inequality at a time. For general mixed integer preprocessing it would be too time consuming to check every pair or even every triple of inequalities in order to detect some model reductions. Since we have the model formulation at hand we can identify those inequalities more quickly (see Section 3.4). The results are given in Table 7. Similar to standard preprocessing the benefit within a branch-and-bound framework is threefold. The LPs are smaller and thus faster to solve, the lower bounds are generally better, and the combinatorial complexity is reduced. The previously unsolved instance `rnd_1c` is now also solved to optimality within the given time limit. For some of the other unsolved instances there are now better lower bounds available, as well in the root node as after the time limit of 3600 seconds is reached.

Lifting coefficients (see Theorems 1–3 in Section 3.5) has an additional effect on the dual bounds and thus on the branch-and-cut solution process, see Table 8. The lower bounds in the root nodes are improved, where the relative improvement (compared to the corresponding values of Table 7) is higher the wider the time windows are. We included our preprocessing technique (higher order variable fixing), hence the number of variables in the preprocessed integer programs actually is identical. Only the number of inequalities has increased, because previously the trivial bounds (1) and (2) on the variables α and τ, respectively, were not counted as inequalities.

The next improvement of the model is the use of the big-M-free reformulation of inequalities (6) that was presented in Section 3.6. In Table 9, one can see the additional effect of the big-M-free reformulation on the overall solution process. (Note that the previous changes to the model are kept.) The dual bounds provided by the root LP are once more improved.

Table 6
Dual bounds with standard presolve

Instance	Variables	Constraints	LP relaxation	Lower bound	Upper bound	Gap (%)	Nodes	Time (s)
`rnd_1a`	193	263	10.0006	19.0052	**19.0052**	0	209	<1
`rnd_1b`	475	546	3.3765	13.0292	**13.0292**	0	51,268	250
`rnd_1c`	648	719	0.0021	9.4199	10.0343	6.5	334,020	3600
`rnd_2a`	313	373	4.1289	14.0101	**14.0101**	0	35,191	169
`rnd_2b`	572	632	0.4047	8.2403	11.0140	33.7	247,788	3600
`rnd_2c`	659	719	0.0016	5.0277	10.0158	99.2	282,765	3600
`rnd_3a`	218	281	7.7555	16.0083	**16.0083**	0	765	4
`rnd_3b`	514	604	1.0017	9.5022	11.0178	15.9	199,157	3600
`rnd_3c`	672	735	0.0021	2.6707	10.0161	275.0	178,594	3600

Table 7
Dual bounds with higher order variable fixing

Instance	Variables	Constraints	LP relaxation	Lower bound	Upper bound	Gap (%)	Nodes	Time
`rnd_1a`	152	222	10.2810	19.0052	**19.0052**	0	80	<1
`rnd_1b`	413	484	3.3769	13.0292	**13.0292**	0	5144	20
`rnd_1c`	563	634	0.0025	10.0343	**10.0343**	0	171,100	514
`rnd_2a`	282	342	4.1294	14.0101	**14.0101**	0	43,318	151
`rnd_2b`	520	580	0.6116	9.2908	11.0139	18.5	654.279	3600
`rnd_2c`	590	650	0.0022	5.0031	10.0159	100.2	325,011	3600
`rnd_3a`	189	251	9.0016	16.0083	**16.0083**	0	537	2
`rnd_3b`	481	544	1.0021	10.1633	11.0176	8.4	291,325	3600
`rnd_3c`	598	661	0.0026	4.1866	10.0190	139.3	202,159	3600

Table 8
Dual bounds with lifting

Instance	Variables	Constraints	LP relaxation	Lower bound	Upper bound	Gap (%)	Nodes	Time
rnd_1a	152	310	12.3151	19.0052	**19.0052**	0	199	<1
rnd_1b	413	599	5.1029	13.0292	**13.0292**	0	4267	40
rnd_1c	563	756	0.9809	10.0343	**10.0343**	0	224,100	3511
rnd_2a	282	450	8.3942	14.0101	**14.0101**	0	7178	46
rnd_2b	520	704	2.0424	8.2255	11.0140	33.9	106,681	3600
rnd_2c	590	774	0.0063	4.6570	10.0163	115.1	117,500	3600
rnd_3a	189	333	11.9703	16.0083	**16.0083**	0	536	3
rnd_3b	481	662	3.7488	9.3286	12.0129	28.8	155,664	3600
rnd_3c	598	781	0.0196	4.1569	10.0166	140.9	79,113	3600

Table 9
Dual bounds with big-M-free reformulation

Instance	Variables	Constraints	LP relaxation	Lower bound	Upper bound	Gap (%)	Nodes	Time
rnd_1a	241	696	12.4504	19.0052	**19.0052**	0	112	1
rnd_1b	764	2008	5.5835	13.0292	**13.0292**	0	3070	120
rnd_1c	1066	2760	2.2844	9.5829	11.0264	15.1	32,105	3600
rnd_2a	492	1295	8.4647	14.0101	**14.0101**	0	15,737	402
rnd_2b	980	2517	2.9120	10.0016	11.0138	10.1	45,600	3600
rnd_2c	1119	2888	1.8844	8.1374	10.0158	23.1	51,636	3600
rnd_3a	314	852	11.9809	16.0083	**16.0083**	0	373	8
rnd_3b	899	2288	4.0882	10.2826	11.0182	7.1	46,426	3600
rnd_3c	1136	2861	2.1297	7.6681	10.0162	30.6	54,800	3600

In particular instances that have the widest time windows benefit most from the big-M-free reformulation. However, the linear programs are larger with respect to the number of constraints and variables. Thus they need more time for solving. After the time limit was reached (for instances that were not solved to optimality before) the number of nodes solved so far is lower than in the other models. On the other hand, the lower bounds from branch-and-bound are better compared to the other models.

The final improvement of the model is the use of k-path cuts. We generate these cuts on demand to strengthen the root relaxation of the model (and keeping all previous changes so far). Further improvements of the dual bound are shown in Table 10. We iteratively apply each of the three separation heuristics until all of them fail to produce new inequalities. The objective function value of the model's root LP relaxation is increased by the k-path cuts. The number of inequalities that are found by each of the three heuristics is also given in Table 10. Then we use branch-and-bound until either optimality or the time limit is reached. Since the k-path cuts are computationally rather expensive we do not generate them within the search tree. For those instances where optimality was proved within the time limit (such as rnd_2b or rnd3_a) much more nodes are now needed. For other instances, where optimality was not reached within the given time limit, much fewer nodes were solved and the gap between lower and upper bound is larger, compared to the results in Table 9. We conclude from this that k-path cuts are only good in improving the root LP, but are computationally less efficient within a branch-

Table 10
Dual bounds with k-path cuts

Instance	Cuts (heur. 1, 2, 3)			LP relaxation	Lower bound	Upper bound	Gap (%)	Nodes	Time
rnd_1a	39	13	10	15.9233	19.0052	**19.0052**	0	61	1
rnd_1b	76	7	5	8.2947	13.0292	**13.0292**	0	6590	301
rnd_1c	38	31	53	3.0716	9.2421	10.0343	8.5	25,832	3600
rnd_2a	29	4	3	9.7089	14.0101	**14.0101**	0	11,487	379
rnd_2b	45	10	13	3.8494	9.3260	11.0144	18.1	29,707	3600
rnd_2c	42	20	23	2.1046	7.2157	10.0158	38.8	33,525	3600
rnd_3a	13	1	2	12.9726	16.0083	**16.0083**	0	783	11
rnd_3b	10	13	12	4.5758	9.5299	12.0126	26.1	27,890	3600
rnd_3c	28	10	9	2.2676	7.0772	10.0199	41.6	36,350	3600

and-bound search. We finally remark that k-path cuts are much stronger on the big-M-free reformulation (55) than on the big-M model (13).

Finally in Tables 11 and 12, we put together the results presented above. The first table shows how the lower bounds from the root LP relaxation benefit from using the various model enhancements. The last table of this section shows the best primal and dual bounds, and optimality gaps obtained with the methods presented above.

Summarizing all results from above, we conclude that the quality of dual bounds from the linear programming relaxation highly depends on the size of the time windows: The wider they are, the more flexibility for the vehicles' schedules they provide, and worser bounds are obtained. The LP relaxation can be improved by strong preprocessing and by adding problem-specific valid inequalities (cutting planes). However, some of the cuts as well as the big-M-free reformulation have a negative influence on the overall performance of the branch-and-bound process by making the linear programs much more difficult to solve or by increasing the number of subproblems (nodes in the branch-and-bound tree). Thus, applying those techniques is not an option to solve larger problem instances.

5.1. Solving the real-world instances

We now turn to the real-world instances. Most of the calculations can be carried out in the same way as above. Due to the larger size of the instances, however, some of the methods (namely, the k-path cuts) are not applied. As a basis we take again the model formulation (13). For a comparison with the current number of buses in the respective county we fix the starting time of schools and trips in the model (13). Solving the remaining problem (i.e., computing x, v, w) yields the results shown in Table 13.

We start again with standard preprocessing techniques: starting time propagation, variable fixing, and big-M-tightening. The results are given in Table 14.

Using the higher order variable fixing we can remove some of the binary variables. The resulting integer programming models are a bit smaller and faster to solve, so that after a given time more nodes are computed within a branch-and-bound search. For most instances (except `Soest`) better lower and upper bounds are found within the given time limit. For one real-world instance, `Wernigerode`, the MIP solver was even able to find a globally optimal solution (see Table 15).

We now improve the lower bound using the coefficient lifting techniques and after that, the big-M-free reformulation. We do not consider k-path cuts here since solving the root LP relaxation using the big-M-free model (55) already takes up to several hours for the larger instances. As one can see in Table 16 the lower bounds are improved by the inclusion of each technique.

Table 11
The hierarchy of root LP relaxation strengthening

Instance	Presolve	High. ord. v.f.	Lifting	Big-M-free	k-path
`rnd_1a`	10.0006	10.2810	12.3151	12.4504	15.9233
`rnd_1b`	3.3765	3.3769	5.1029	5.5835	8.2947
`rnd_1c`	0.0021	0.0025	0.9809	2.2844	3.0716
`rnd_2a`	4.1289	4.1294	8.3942	8.4647	9.7089
`rnd_2b`	0.4047	0.6116	2.0424	2.9120	3.8494
`rnd_2c`	0.0016	0.0022	0.0063	1.8844	2.1046
`rnd_3a`	7.7555	9.0016	11.9703	11.9809	12.9726
`rnd_3b`	1.0017	1.0021	3.7488	4.0882	4.5758
`rnd_3c`	0.0021	0.0026	0.0196	2.1297	2.2676

Table 12
Best primal and dual bounds

Instance	Lower bound	Upper bound	Gap (%)
`rnd_1a`	19.0052	**19.0052**	0
`rnd_1b`	13.0292	**13.0292**	0
`rnd_1c`	10.0343	**10.0343**	0
`rnd_2a`	14.0101	**14.0101**	0
`rnd_2b`	10.0016	11.0138	10.1
`rnd_2c`	8.1374	10.0158	23.1
`rnd_3a`	16.0083	**16.0083**	0
`rnd_3b`	10.2826	11.0176	7.1
`rnd_3c`	7.6681	10.0161	30.6

Table 13
Current number of buses and deadhead trips

Instance	Objective
Demmin	82.0423
Steinfurt	226.0711
Soest	90.0488
Wernigerode	43.0093
Guetersloh	176.0725

Table 14
Dual bounds, standard presolve

Instance	Variables	Constraints	LP relaxation	Lower bound	Upper bound	Gap (%)	Nodes	Time
Demmin	28,566	23,549	15.1263	39.1513	66.0553	68.7	18,100	3600
Steinfurt	71,328	65,885	31.6287	88.9567	187.1361	110.4	240	3600
Soest	14,620	14,824	6.3873	20.1343	70.0864	248.1	11,700	3600
Wernigerode	6826	4,931	21.3498	37.3984	38.0198	1.6	174,542	3600
Guetersloh	57,718	46,406	45.1726	85.6738	140.1084	63.5	3451	3600

Table 15
Dual bounds with higher order variable fixing

Instance	Variables	Constraints	LP relaxation	Lower bound	Upper bound	Gap (%)	Nodes	Time
Demmin	28,188	23,170	15.1341	39.1369	65.0585	66.2	17,211	3600
Steinfurt	69,992	64,547	31.9827	88.4651	182.1380	105.8	440	3600
Soest	14,222	14,426	6.3930	21.5686	73.0879	238.8	16,700	3600
Wernigerode	6768	4873	23.3905	38.0198	**38.0198**	0	62,658	1659
Guetersloh	56,922	45,604	48.7121	86.1151	139.1114	61.5	3172	3600

Table 16
The hierarchy of root LP relaxation strengthening

Instance	Presolved	High. ord. v.f.	Lifting	Big-M-free
Demmin	15.1263	15.1341	22.7273	25.4147
Steinfurt	31.6287	31.9827	70.6715	75.1771
Soest	6.3873	6.3930	15.0733	18.5776
Wernigerode	21.3498	23.3905	28.0122	28.0409
Guetersloh	45.1726	48.7121	74.8144	77.1386

6. Conclusions

In this article, we gave an integer programming formulation for the integrated optimization of school starting times and public transport. The development of this model was driven by the real-world demand of counties in rural areas, where pupils are the largest group of customers, and most of the schools start currently at the same time. For the solution of the model we used branch-and-cut techniques. To reduce the computation times we analyzed the mathematical model and presented several methods to strengthen the formulation. Their computational benefit was demonstrated on the basis of several artificial and real-world problem instances.

It turned out that using integrated optimization of school starting times and public transport it is possible to reduce the number of deployed buses by 10–25% (or 5–50 in absolute values). The reduction of a single bus is worth 30,000 Euro of tax allowance for the county adminstration, or 200,000 Euro of operating costs for the bus company. For a single county, the whole optimization can save up to 1 Mio. Euro – year by year. Extrapolated to all of 323 counties in Germany, a lot of public money could be saved by a thoroughly application of the presented methods.

References

[1] N. Ascheuer, M. Fischetti, M. Grötschel, A polyhedral study of the asymmetric traveling salesman problem with time windows, Networks 36 (2) (2000) 69–79.
[2] N. Ascheuer, M. Fischetti, M. Grötschel, Solving the asymmetric travelling salesman problem with time windows by branch-and-cut, Mathematical Programming 90 (2000) 475–506.

[3] L. Bodin, L. Berman, Routing and scheduling of school buses by computer, Transportation Science 13 (2) (1979) 113–129.
[4] R.L. Bowerman, G.B. Hall, P.H. Calamai, A multi-objective optimisation approach to school bus routing problems, Transportation Research Part A 28 (5) (1995) 107–123.
[5] J. Braca, J. Bramel, B. Posner, D. Simchi-Levi, A computerized approach to the New York City school bus routing problem, IIE Transactions 29 (1997) 693–702.
[6] A. Corberan, E. Fernandez, M. Laguna, R. Marti, Heuristic solutions to the problem of routing school buses with multiple objectives, Journal of the OR Society 53 (4) (2002) 427–435.
[7] J.-F. Cordeau, G. Desaulniers, J. Desrosiers, M.M. Solomon, F. Soumis, VRP with time windows, in: P. Toth, D. Vigo (Eds.), The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, 2002, pp. 157–193.
[8] G. Desaulniers, J. Lavigne, F. Soumis, Multi-depot vehicle scheduling with time windows and waiting costs, European Journal of Operational Research 111 (1998) 479–494.
[9] M. Desrochers, G. Laporte, Improvements and extensions to the Miller–Tucker–Zemlin subtour elimination constraints, Operations Research Letters 10 (1991) 27–36.
[10] A. Fügenschuh, The Integrated Optimization of School Starting Times and Public Transport, Ph.D. Thesis, Logos Verlag, Berlin, 2005, 175 pp.
[11] A. Fügenschuh, A. Martin, Computational integer programming and cutting planes, in: K. Aardal, G. Nemhauser, R. Weissmantel (Eds.), Handbook on Discrete Optimization. Series Handbooks in Operations Research and Management Science, Elsevier, 2005, pp. 69–122.
[12] ILOG CPLEX Division, 889 Alder Avenue, Suite 200, Incline Village, NV 89451, USA, information available at URL http://www.cplex.com.
[13] I. Kara, G. Laporte, T. Bektas, A Note on the Lifted Miller-Tucker-Zemlin Subtour Elimination Constraints for the Capacitated Vehicle Routing Problem, Technical Report, Les Cahiers du GERAD, 2003, G-2003-12.
[14] H. Keller, W. Müller, Optimierung des Schülerverkehrs durch gemischt ganzzahlige Programmierung, Zeitschrift für Operations Research B 23 (1979) 105–122 (in German).
[15] N. Kohl, J. Desrosiers, O.B.G. Madsen, M.M. Solomon, F. Soumis, 2-Path cuts for the vehicle routing problem with time windows, Transportation Science 33 (1999) 101–116.
[16] M. Spada, Sur la Planifacation de Tournees de Vehicules Scolaires, Ph.D. Thesis, Lausanne, EPFL, 2004, 115 pp. (in French).
[17] F. Maffioli, A. Sciomachen, A mixed-integer model for solving ordering problems with side constraints, Annals of Operations Research 69 (1997) 277–297.
[18] M.W. Padberg, A note on zero-one programming, Operations Research 23 (4) (1975) 833–837.
[19] P. Stöveken, Wirtschaftlicherer Schulverkehr: ÖPNV-Optimierung mit erfolgsabhängiger Honorierung, Der Nahverkehr 3 (2000) 65–68 (in German).
[20] C.A. van Eijl, A Polyhedral Approach to the Delivery Man Problem, Technical Report 95-19, Department of Mathematics and Computer Science, Eindhoven University of Technology, 1995.
[21] Wikipedia, the Free Encyclopedia, 2006, online available at URL http://www.wikipedia.org.
[22] L.A. Wolsey, Faces of linear inequalities in 0–1 variables, Mathematical Programming 8 (1975) 165–178.