



Revisiting the richness of integrated vehicle and crew scheduling

Liping Ge¹ · Natalia Kliewer² · Abtin Nourmohammadzadeh³ ·
Stefan Voß³ · Lin Xie¹

Accepted: 27 January 2022
© The Author(s) 2022

Abstract

The last decades have seen a considerable move forward regarding integrated vehicle and crew scheduling in various realms (airline industry, public transport). With the continuous improvement of information and communication technology as well as general solvers it has become possible to formulate more and more rich versions of these problems. In public transport, issues like rostering, delay propagation or days-off patterns have become part of these integrated problems. In this paper we aim to revisit an earlier formulation incorporating days-off patterns and investigate whether solvability with standard solvers has now become possible and to which extent the incorporation of other aspects can make the problem setting more rich and still keep the possible solvability in mind. This includes especially issues like delay propagation where in public transport delay propagation usually refers to secondary delays following a (primary) disturbance. Moreover, we investigate a robust version to support the claim that added richness is possible. Numerical results are provided to underline the envisaged advances.

✉ Stefan Voß
stefan.voss@uni-hamburg.de

Liping Ge
liping.ge@leuphana.de

Natalia Kliewer
natalia.kliewer@fu-berlin.de

Abtin Nourmohammadzadeh
abtin.nourmohammadzadeh@uni-hamburg.de

Lin Xie
lin.xie@leuphana.de

¹ Leuphana University Lüneburg, Institute of Information Systems, Universitätsallee 1, 21335 Lüneburg, Germany

² Free University Berlin, Information Systems, Garystr. 21, 14195 Berlin, Germany

³ University of Hamburg, Institute of Information Systems, Von-Melle-Park 5, 20146 Hamburg, Germany

1 Introduction

Operations research is playing a considerable role in solving real-world problems in public transportation with specific problem settings becoming more and more integrated. A major breakthrough regarding visibility outside of the public transportation domain was the success of these approaches when being judged at the Edelman award more than 15 years ago (see Abbink et al. 2005). Ever since, more and more details have made problem settings richer and richer. While the connotation of rich problems is widely known in the vehicle-routing domain (cf. Hartl et al. 2006), we can also envisage them in public transport. Starting from the classical vehicle and crew scheduling problems, various extensions have been tackled in literature like legal constraints, fairness constraints, company-related governance rules etc. Examples that were striking our interest over time include rostering and days-off patterns in Mesquita et al. (2013) or the consideration of fixed buffer times and delay propagation in Amberg et al. (2019).

Over the years, we also saw major achievements of solution methodology in solving hard optimization problems in operations research and recently, we can find a wealth of general purpose solvers¹ being ready to solve problems, once they have been “mipped”. That is, following Fischetti et al. (2009), it pays to formulate a problem by means of a mathematical model like a mixed-integer programming (MIP) model and then solve it using related software (including those general solvers as well as especially tailored matheuristics and metaheuristics). Moreover, even classical methods like Benders decomposition can now be enhanced in a matheuristic fashion and considerably be improved (see, e.g., Caserta and Voß 2021).

In the light of all these improvements, we are asking the following research questions:

- **R1:** Is it possible to solve integrated problems in public transport, that had to be tackled by means of specialized heuristics years ago due to their inherent problem complexity, by means of currently available standard solvers and, if so, which instance sizes are to be solved in time limits deemed practical?
- **R2:** Is it possible to integrate additional problem features in given vehicle and crew scheduling problems to make them richer, without fully discarding the solvability by means of standard solvers?

To answer these research questions, we go back to earlier solution approaches in integrated vehicle and crew scheduling problems. After briefly surveying this problem domain in Sect. 2, we single out two approaches from the related literature in Sect. 3, namely the above mentioned references of Mesquita et al. (2013) and Amberg et al. (2019). After having investigated the settings of Mesquita et al. (2013)

¹ See, e.g., https://www.ibm.com/support/knowledgecenter/SSSA5P_12.10.0/COS_KC_home.html and <http://www.gurobi.com/products/gurobi-optimizer/gurobi-overview>. (last call 23 Dec 2021) As modeling environment with the option to call a solver like the CPLEX one on top, we use the GAMS environment (see GAMS Development Corporation 2020).

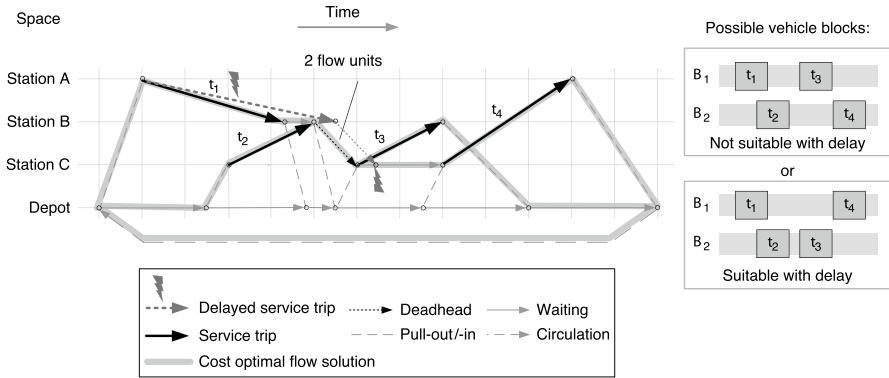


Fig. 1 Time-space network example with four service trips from Amberg et al. (2019, p. 95)

first, Sects. 3.2 and 3.3 investigate problem extensions to make those already rich problems a little richer and ask the same research questions again. The final section concludes and offers ideas for future research.

2 Literature review

Planning problems in public transport are usually classified into different types of problems depending on which impact they have, i.e., strategic, tactical and operational. We assume that the reader is acquainted with this in general terms (see, e.g., Ceder 2015; Daduna and Voß 2000; Desaulniers and Hickman 2007; Vuchic 2005).

First, we consider issues in public transport with a focus on vehicle and crew scheduling. Then we turn towards more general issues in mixed-integer programming.

2.1 Integrated vehicle and crew scheduling

According to Desaulniers and Hickman (2007, p. 104/105), the integrated vehicle and duty scheduling problem (IVDSP) is defined as follows. “Given a set of timetabled trips and a fleet of vehicles assigned to several depots, find minimum-cost vehicle blocks and valid driver duties such that each active trip is covered by one block, each active trip segment is covered by one duty, and each deadhead, pull-in, and pull-out trip (hereafter called an inactive trip) used in the vehicle schedule is also covered by one duty”. Assumptions usually include that blocks need to start and end at the same depot, several work rules have to be considered, drivers are originating also from one depot, and possibly many additional ones. Figure 1 aims to clarify this situation in an extended setting for a single depot and four trips t_1, t_2, t_3, t_4 . Assigning duties to drivers is usually called rostering.

Vehicle and crew scheduling in public transport are usually solved as tactical and operational problems and this is often done in a hierarchical fashion, mostly due to the inherent difficulty of the problems (see, e.g., Freling et al. 1999; Desaulniers and

Hickman 2007; Perumal et al. 2020) for surveys over time and the embedding of this area into operations research and related planning and operations.

Usually, vehicle and crew scheduling have been solved in a hierarchical or an integrated way. The hierarchical way can also be called sequential, where vehicle schedules are specified first before crew schedules are determined. That is, for existing lines, trips are defined for which vehicles are scheduled and then crews are assigned to duties covering these trips. An additional common distinction accounts for distinguishing the single-depot and the multi-depot case. Early references for integrated problem settings are some PhD-theses on the topic (see, e.g., Freling 1997) and related papers (see, e.g., Freling et al. 2001; Haase et al. 2001; Huisman et al. 2005).

From the recent survey of Perumal et al. (2020) we can make an interesting observation which is that in the 10-year period from 2009/2010 there are not so many works any more on the integrated vehicle and crew scheduling problem. Exceptions are to some extent those that try to emphasize practical settings as, e.g., Mesquita et al. (2013); Er-Rbib et al. (2021b) for rostering (both with days-off patterns). Only recently, the interest in these integrated problems became rejuvenated with the focus of making the problem settings richer. In this respect, a few issues of interest within public transport refer to “complications” of the problem settings. One of these issues relates to the inclusion of buffer times. This can be done upfront by simple inclusion of extra time, often known as schedule padding (see, e.g., Wessel and Widener 2017), or by means of calculations with fixed or variable buffer times (see, e.g., Swierstra et al. 2017). Examples for the latter include delay propagation by Amberg et al. (2019). New ideas and attempts have been motivated especially by new technologies like electric buses (cf. Perumal et al. 2020) or by means of incorporating issues regarding disturbances and robustness (cf. Borndörfer et al. 2010; Ge et al. 2020). Fairness settings regarding workload balancing are considered (e.g., by Xie and Suhl 2015; Er-Rbib et al. 2021a).

While integrated vehicle and crew scheduling is a topic with a great interest in public transport, it can also be found in related areas. Starting from the vehicle routing problem and opening up also to logistics, we can see an explosion of various integrated settings. In logistics the integration may include the joint movement of various transport modes (like a vehicle with a driver and an unmanned drone); see, e.g., Otto et al. (2018) for a survey on this type of problems. Starting with the classical vehicle routing problem, Lam et al. (2020) relax the usual assumption that only one crew operates a vehicle in that problem setting. With a focus on humanitarian and military logistics, the authors formulate in mathematical terms combining MIP approaches with constraint programming. In the same spirit, we can find crew rostering problems with added richness in the airline industry (see, e.g., Doi et al. 2018). Integrated vehicle and crew scheduling in the maritime industry is rather scarce (where vehicles are assumed to be vessels). For an exception see Ang Pik Yoke et al. (2021). In all these cases data availability seems a major issue; for a recent survey regarding public transport see Ge et al. (2021).

Mathematical modeling attempts for integrated vehicle and crew scheduling are mostly utilizing either a network-flow formulation or the use of the set partitioning or set covering problem as an underlying problem structure. Most early approaches

for the exact solution of integrated problem settings were based on these settings by applying branch-and-bound or branch-and-cut approaches using multi-commodity flow or set partitioning- and set covering-based modeling approaches (Desaulniers and Hickman 2007; Perumal et al. 2020). Advances in this respect are still observed over time (see, e.g., Tahir et al. 2019), who investigate a set partitioning-based approach for vehicle scheduling. Using dynamic programming and an adjacency-based algorithm, recently Himmich et al. (2020) propose a primal column generation framework. Moreover, various types of heuristics and metaheuristics have been tried, including even especially tailored Benders-based heuristics (like the one of Mesquita et al. 2013).

As an interlude we should state that typical problem sizes in literature are largely varying. Shen and Xia (2009) solve data from the Beijing Bus group using an iterative sequential heuristic algorithm and they can find feasible solutions for instances up to 107 buses and 164 duties. Mesquita et al. (2011) use data from a bus company in Lisbon to demonstrate their preemptive goal programming-based heuristic approach while they solve the integrated problem two years later (Mesquita et al. 2013) with a Benders decomposition approach. All three papers evaluate their proposed algorithms on real-world instances, but the timetabled trips are limited to 238. To be more specific, we like to re-iterate the details of the instance sizes going up to 238 trips. The first set of data comes from different bus lines from Lisbon, Portugal, with 122, 168, 224, 226 and 238 timetabled trips for weekdays and vehicles from four depots. The second set comes from Porto, Portugal, with 108, 156, 250 and 280 timetabled trips for weekdays and vehicles from one depot. The results reported in Mesquita et al. (2013) also incorporate those for selected data from the repository of Huisman (2007) (the repository includes instances with up to 400 trips). The data used incorporate start and end times as well as start and end locations of all trips for instances with 80 and 200 trips, respectively, involving four depots (for weekdays). Weekend trips were randomly selected, with a probability of 0.4 from the respective non-weekend daily trip sets. Below, we shall show that we can solve even larger instances to optimality.

Our approach in this paper is to revisit earlier formulations from literature. That is, we repeat a given formulation, focus on its solvability under today's standard solver conditions, and then show that several as yet unexplored extensions to that formulation are within the same range regarding solvability.

Looking at possible extensions from literature, we like to point to very recent ones that are difficult to integrate even with today's techniques. As an example, consider the work of Er-Rbib et al. (2021b). For a given set of predefined duties and groups of drivers, they define the duty assignment problem with group-based driver preferences as the problem of building rosters that cover all the duties over a predetermined cyclic horizon while respecting a set of rules (hard constraints), balancing the workload between the drivers and satisfying as much as possible the driver preferences (soft constraints). In a sense this may be referred to as a problem setting seeking fairness among the drivers. The problem also considers fixed days off (which relates it to the settings of Mesquita et al. 2013). Their mathematical programming formulation allows to optimally solve instances with up to 124 drivers and 490 duties while keeping a near-optimal workload balance. Larger

instances with up to 333 drivers and 1509 duties are treated with a newly developed matheuristic, though without guaranteeing optimality for the related results. A different set of constraints to be integrated stems from Perumal et al. (2021). Given a set of timetabled bus trips, they search for a driver schedule covering a given set of trips obeying various labor union regulations (especially referring to breaks for the drivers). In addition, drivers possibly need to travel by separate vehicles (called staff cars) between bus stops to have their breaks. The simultaneous scheduling of drivers and staff cars for the drivers is considered as the driver scheduling problem with staff cars. The authors consider several instances of different sizes from a number of European service providers (with a nondisclosure agreement so that the detailed data are not available to readers). The number of trips varies from 43 to 1926 and the number of staff cars ranging within 0 to 6, but they can be as large as 15 for one of the instances. While the small instances can be solved to optimality using the MIP model, most larger ones are out of reach for the model and a matheuristic is used (using the mathematical programming formulation and an adaptive large neighbourhood search).

2.2 Mixed-integer programming and replication studies

Many of the advances in mathematical programming utilized in general, i.e., not only in integrated vehicle and crew scheduling, come from the advances of information and communication technologies as well as general purpose solvers like CPLEX and Gurobi. This is also in line with the advances in general MIP solving. An entry point into these advances may be given using Gleixner et al. (2021) and Mittelmann (2020). A comparative analysis of different solvers can be found, e.g., in Anand et al. (2017) and within the links provided in Mittelmann (2020).

Next, we like to mention an area that has not yet comprehensively been combined with public transport research (if at all), namely replication studies. First of all, for many mathematical programming formulations in combinatorial optimization, like for the traveling salesman problem, it is well established to re-use existing formulations to get new insights on the specific problem under consideration. Within the social sciences, e.g., Hüffmeier et al. (2016) argue that replications may have a lot to offer beyond what is already known, especially if an improved conceptualization is available. This may be appended if new or improved technology or algorithmic advances have appeared over time. Related arguments can be found in various areas including management (see, e.g., Block and Kuckertz 2018; Boylan 2016). Putting this into perspective, we can argue that modern information and communication technology, standard solver technology as well as the upcoming use of matheuristics has made the case also for replication studies, e.g. using existing mathematical programming formulations in public transport. Even in the field of software engineering and metaheuristics this is mostly unexplored (see, e.g., Kendall et al. 2016; Swan et al. 2022). A starting point within software engineering could even be the idea of back-to-back testing (see, e.g., Jörges and Steffen 2014).

To the best of our knowledge, a general framework for replication studies in mathematical programming does not yet exist. There are a few thoughts that may be

used to provide some pointers to related literature. Coming from software engineering and algorithm design as well as heuristics or metaheuristics, re-use needs to be preplanned (see, e.g., Fink and Voß 2002 with a focus on metaheuristics class libraries). In a sense, this applies to solvers as well as methods themselves. Over time, many algorithmic approaches have been published without always being rigorous or standing the test of novelty (see, e.g., Sörensen et al. 2019 for a specific example and de Armas et al. 2021 for the proposal of a template in the area of metaheuristics). Differently, this topic relates to replicating studies performed on developed models in mathematical programming, discrete event simulation, etc. (see, e.g., the discussion in Taylor et al. 2018). It seems still seldom that a repository is set up with the underlying data and models as well as algorithms being provided for readers. And if it is (like in case of a heuristic optimization framework from the above-mentioned Fink and Voß 2002), then one often finds counter-arguments regarding its use related to a “not-invented-here syndrome” (see, e.g., Nissen 2018).

As this has not yet been extensively studied, we just provide some hint on where this could go, e.g., in the field of metaheuristics and evolutionary programming like López-Ibáñez et al. (2021), Swan et al. (2022) and de Armas et al. (2021). Within mathematical programming the arguments used when setting up the MIPLIB library, including quite a few instances from our scheduling domain, should be used as a starting point (see Koch et al. 2011; Gleixner et al. 2021) even if some counter-measures exist (see Mittelmann 2020). A clear-cut standpoint might be to provide at least information about the computing environment, if separate programming is needed, the programming language, information about the used software or solver, the model itself, the used data, etc.

Supporting arguments can be found in the works around erraticism, i.e., replicating existing studies with different seeds for random number generators as proposed by Fischetti and Monaci (2014). The idea is to replicate numerical results with the same model and just different seed generators. Extending this, Voß and Lalla-Ruiz (2016) reformulate the multiple-choice multidimensional knapsack problem as a generalized set partitioning problem with the results of obtaining various new best known results for one of the most studied problems in combinatorial optimization. And, just to support the case, they even show that the heuristic cut generation within CPLEX and Gurobi may lead to different new best results. Extending this may lead to the idea of utilizing redundant constraints to strategically influence the success of the solver, as shown in Lalla-Ruiz and Voß (2016). On a similar scale, one may also consider the same type of model, the same type of algorithm and have just some slight modification. As an example with an application towards integrated vehicle and crew scheduling we refer to a modified branching strategy within branch-and-bound proposed in Borndörfer et al. (2013), named rapid branching.

3 Selected problem setting(s) and solvability

As we have seen in the previous section, there are quite a few works emphasizing integrated vehicle and crew scheduling. As examples incorporating a certain type of richness, we resort to the work of Mesquita et al. (2013). Regarding our research

questions it seems most suitable due to its age and the related innovation of that time (applying an advanced Benders heuristic based on an appropriate mathematical modeling approach). That is, the time between their exposition and now resorts in a perfect way towards our research questions. Moreover, in the spirit of considering (“revisiting”) a certain type of richness, this work incorporates some legal constraints as well as days-off patterns. In addition, buffer times as well as robustness issues are treated.

3.1 The model of Mesquita et al.

Let us consider the integrated vehicle and crew scheduling or rostering problem (i.e. minimum cost vehicle and daily crew schedules that cover all timetabled trips and a minimum cost roster covering all daily crew duties according to a pre-defined days-off pattern) as provided in Mesquita et al. (2013), and let us start with some notation following that reference. Firstly, we use N^h to indicate the set of trips to be performed on a specific day h with a planning horizon of H . Secondly, binary variables z_{ij}^{dh} indicate whether a vehicle from a specific depot d (out of a given set of depots \mathcal{D}) performs trip i immediately before trip j . Moreover, I^h indicates the set of pairs of compatible trips (i, j) , i.e., any two trips i and j which can be performed immediately one after another. These are regular trips as well as those coming from a depot (called pull-out trips; see also Fig. 1) or going into a depot (called pull-in trips). The cost values to be considered for trips i and j to be performed in immediate succession are c_{ij}^d . Note that Mesquita et al. (2013) do not distinguish these values for specific days (like weekdays versus weekends or holidays). Moreover, pull-in trips are implicitly considered due to the nature of the used flow conservation constraints (see (3) in the model below). Additional data include v_d as the number of vehicles available at depot $d \in \mathcal{D}$. Moreover, we define a set of pull-in trips as \mathcal{D}' .

L_{ij}^h denotes the set of crew duties possibly covering task (i, j) on day h , where this is a subset of the set L^h being the set of all crew duties of a specific day h . Each possible crew duty l incurs a cost value of e_l . Variables w_l^h are related variables indicating whether crew duty l is selected on day h . Each driver m belongs to a set of drivers M . With that, binary variables y_l^{mh} may be defined indicating whether driver m performs a certain crew duty l on day h . To perform a duty, a driver m must be available and scheduled according to a certain schedule $s \in S$ which defines his / her availability during certain days, hours, etc. This even may incorporate certain duties that are differentiated from regular duties, i.e., short duties of at most 5 h without break and long duties with overtime beyond 9 h. That is, S indicates the set of possible schedules or schedule variations (like a regular day shift on a weekday, on a weekend etc.). Binary variables x_s^m together with cost values r^m are used to display this. In Mesquita et al. (2013) this is also emphasized by means of a binary parameter a_s^h stating whether a certain day h is included in schedule (or schedule type) s . Variables η_T and η_O are used to account for the used numbers of short and long duties, respectively, incurring penalties λ_T and λ_O if they are used. Short duties are

those up to 5 h and long duties are those of more than 9 h;² all others are called normal duties. Note that these numbers are bounded from above by means of the data of each specific problem instance. Set L^h may have related subsets defined, L_T^h for short duties, L_O^h for long duties, and L_N^h for those normal duties without being specified as being either short or long. Without having to pay extra costs, one may also distinguish different duties starting in the first or the latter part of a day, say, a set L_E^h indicating those which start any time in the morning up to 3:30 pm (they, e.g., need to incorporate a lunch break for drivers) and a set L_A^h starting at 3:30 pm or later.

To ease with the notation, we repeat the complete notation as follows. Note that we already include some notation that is going to be used later in the attempt to provide a formulation incorporating additional means of richness (regarding delay propagation).

| Parameters | |
|--|--|
| c_{ij}^d | The cost of the deadhead trip from i to j |
| e_l | The duty cost of l |
| $L^h, L_E^h, L_A^h, L_T^h, L_N^h, L_O^h$ | The set of crew duties, the set of early duties, the set of late duties, the set of short duties, the set of normal duties, the set of long duties |
| $L_{ij}^h \subset L^h$ | The set of crew duties covering task (i,j) on day h |
| M | The set of drivers |
| D | The set of depots |
| N^h | The set of trips to be performed on day h |
| v_d | The number of vehicles available at d |
| a_s^h | =1 if h is a workday on schedule s , 0 otherwise |
| r^m | The assignment cost of driver m to a schedule |
| σ_k^d | The delay associated with duty k if it is done with the vehicles of depot d |
| λ_T | The penalty for short driver duties |
| λ_O | The penalty for long driver duties |
| α_V | The coefficient for the effect of delay propagation in terms of vehicle tasks |
| α_C | The coefficient for the effect of delay propagation in terms of crew duties |
| TK | The set of all duties in the chronological order of their execution time, $TK = \{1, 2, \dots, TK \}$ |
| K_{ij}^C | The set of crew duties that can serve (i, j) , $K_{ij}^C \subset TK$ |
| K_{ij}^V | The set of vehicle duties that cover the trip associated with (i, j) , $K_{ij}^V \subset TK$ |
| ED_k | Expected delay of duty k |
| BT_k | The buffer time before the execution time of k |

² This is motivated by circumstances from the real-world cases as presented in Mesquita et al. (2013).

Variables

| | |
|----------------------------|--|
| z_{ij}^{dh} | =1 if a vehicle from depot d performs trips i and j in sequence on day h |
| z_{dj}^{dh}, z_{id}^{dh} | The pull-out from d to trip j and the pull-in from i to depot d |
| w_l^h | If crew duty l is selected on day h |
| x_s^m | If driver m is assigned to schedule s |
| y_l^{mh} | If driver m performs crew duty l on day h |
| η_T, η_O | The maximum number of short, the maximum number of long duties assigned to a driver during H |
| DP_k | Delay propagation up to duty k |
| ATD_k^d | The actual total delay of duty k regarding the vehicles of depot d |
| P_k^d | Binary variable=1, if crew duty k is selected (using depot d) |
| q_k^d | Binary variable=1, if vehicle duty k is selected (using depot d) |

Now, we can model as follows:

$$\min \sum_{h \in H} \left\{ \sum_{d \in \mathcal{D}} \sum_{i:(i,j) \in I^h} c_{ij}^{d, dh} + \sum_{l \in L^h} e_l w_l^h \right\} + \sum_{m \in M} \sum_{s \in S} r^m x_s^m + \lambda_T \eta_T + \lambda_O \eta_O \quad (1)$$

$$\text{s.t.} \quad \sum_{d \in \mathcal{D}} \sum_{i:(i,j) \in I^h} z_{ij}^{dh} = 1 \quad \forall j \in N^h - \mathcal{D}, \forall h \in H \quad (2)$$

$$\sum_{i:(i,j) \in I^h} z_{ij}^{dh} - \sum_{i:(j,i) \in I^h} z_{ji}^{dh} = 0 \quad \forall j \in N^h, \forall d \in \mathcal{D}, \forall h \in H \quad (3)$$

$$\sum_{i \in N^h} z_{di}^{dh} \leq v_d \quad \forall d \in \mathcal{D}, \forall h \in H \quad (4)$$

$$\sum_{l \in L_{ij}^h} w_l^h - \sum_{d \in \mathcal{D}} z_{ij}^{dh} \geq 0 \quad \forall (i,j) \in I^h, \forall h \in H \quad (5)$$

$$\sum_{m \in M} y_l^{mh} - w_l^h = 0 \quad \forall l \in L^h, \forall h \in H \quad (6)$$

$$\sum_{s \in S} x_s^m \leq 1 \quad \forall m \in M \quad (7)$$

$$\sum_{l \in L^h} y_l^{mh} - \sum_{s \in S} \alpha_s^h x_s^m \leq 0 \quad \forall m \in M, \forall h \in H \quad (8)$$

Table 1 Input data generation

| | |
|------------------------|---|
| c_{ij}^d | U[1,5] |
| e_t | U[5,15] |
| v_d | U[10,30] for small instances with enough depots; otherwise = $\frac{\text{the number of vehicles}}{\text{the number of depots}}$ |
| r^m | U[10,20] |
| d_s^h | =1 if h is a workday on schedule s |
| λ_T, λ_O | =2 |

$$\sum_{l \in L_f^h} y_l^{mh} + \sum_{l \in L_g^{h-1}} y_l^{m(h-1)} \leq 1 \quad \forall m \in M, \forall h \in H - \{1\}, f \neq g \in \{E, A\} \tag{9}$$

$$\sum_{h \in H} \sum_{l \in L_t^h} y_l^{mh} - \eta_t \leq 0 \quad \forall m \in M, t \in T, O \tag{10}$$

$$z_{ij}^{dh} \in \{0, 1\} \quad \forall (i, j) \in I^h, \forall d \in \mathcal{D}, \forall h \in H \tag{11}$$

$$w_l^h \in \{0, 1\} \quad \forall l \in L^h, \forall h \in H \tag{12}$$

$$y_l^{mh} \in \{0, 1\} \quad \forall l \in L^h, \forall m \in M, \forall h \in H \tag{13}$$

$$x_s^m \in \{0, 1\} \quad \forall s \in \mathcal{S}, \forall m \in M \tag{14}$$

$$\eta_T, \eta_O \in \mathbb{N}_0 \tag{15}$$

The objective function (1) measures the quality of the solution in terms of vehicle and driver costs and roster balancing (i.e., the third term in the objective is the total assignment cost of drivers to schedules), as well as penalties for short and long duties. Explaining the constraints of the model could start from clarifying that Equalities (2) and (3) indicate a vehicle scheduling problem where the first set of equalities states that each timetabled trip has to be done exactly once by means of choosing one of the vehicles from the same depot having done an immediately preceding trip. Each depot has only a limited number of vehicles available, indicated in (4). Constraints (5) guarantee that each task in a vehicle schedule is covered by at least one crew duty, i.e., here we have the coupling of vehicle and crew duty variables. Equalities (6) ensure that each crew duty of a solution to the problem must be assigned to a driver. Constraints (7) guarantee that each driver is assigned to at most one certain schedule or a certain service. The model also incorporates a few important coupling constraints. Constraints (8) are used to link the assignment of a crew

Fig. 2 CPU-times for modified numbers of duties

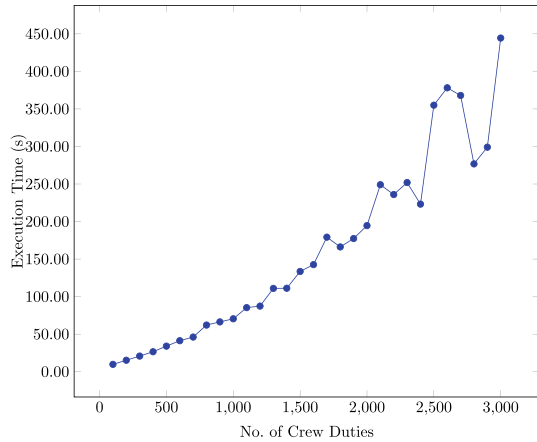
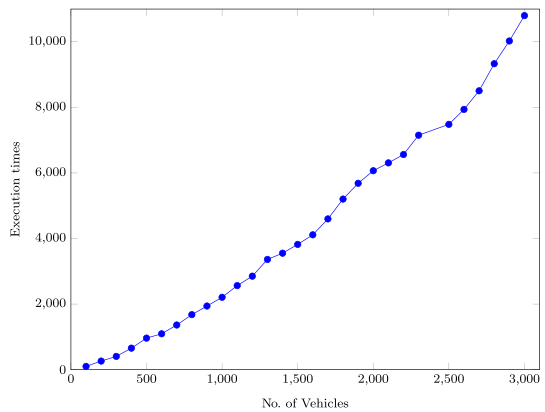


Fig. 3 Execution times (s) for modified numbers of vehicles



duty and a specific schedule (type) assigned to a driver. Constraints (9) are intended to forbid that an early duty follows immediately after a late duty.

Regarding solvability of the model by means of common solvers, we proceed as follows. The model is programmed in GAMS (with CPLEX as the underlying solver) and for the first set of experiments, its input data is randomly generated there. The input generation rules are summarized in Table 1. The computing environment uses a standard PC / laptop (Intel(R) CoreTM i7-6700HQ with 2.60 GHz and 32 GB RAM, 64-Bit-operating system). All models and data of this paper are available from the authors upon request.

The set of crew duties (L^h), the set of early duties (L_E^h), the set of late duties (L_A^h), the set of short duties (L_T^h), the set of normal duties (L_N^h), and the set of long duties (L_O^h) are randomly chosen from the set of all duties.

Instances with different numbers of planning days, duties, depots, trips and drivers are generated as indicated in the next few figures. To generate the number of available vehicles at each depot (v_d), for small instances with enough depots, v_d is

Fig. 4 Objective function values for increasing numbers of duties

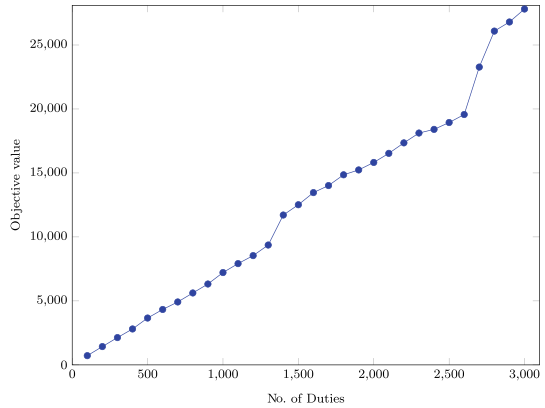
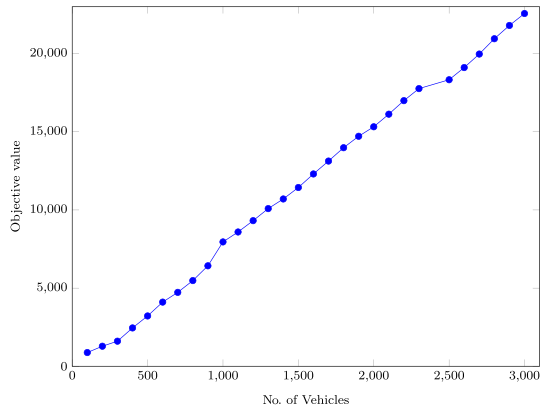


Fig. 5 Objective function values for modified numbers of vehicles



randomly chosen in the interval $[10, 30]$. However, if the instance does not contain enough depots to use this method, then the vehicles are equally divided between them. Here, instances with very large numbers of vehicles are built to examine the effect of this factor on the solver ability or the computation time. Figure 2 shows the execution time of solving the problem with CPLEX. It seems evident that the elapsed time increases drastically as the set of crew duties is enlarged. Likewise, Fig. 3 illustrates the execution time versus the number of vehicles. The resulting objective function values with increasing numbers of crew duties and vehicles are depicted in Figs. 4 and 5, respectively.

In the following, the model is once again solved with its input data provided by Huisman (2007) (regarding the vehicle and crew scheduling part). This is in line with the data generation for some of the instances generated in Mesquita et al. (2013).³ The

³ We should note that the original data from Mesquita et al. (2013) were not available to us, especially due to a non-disclosure agreement regarding the real-world data from Porto and Lisbon, Portugal.

Fig. 6 CPU-times for modified numbers of depots

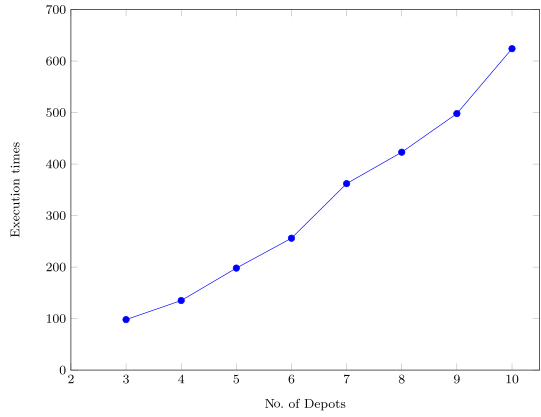


Fig. 7 CPU-times for modified numbers of trips

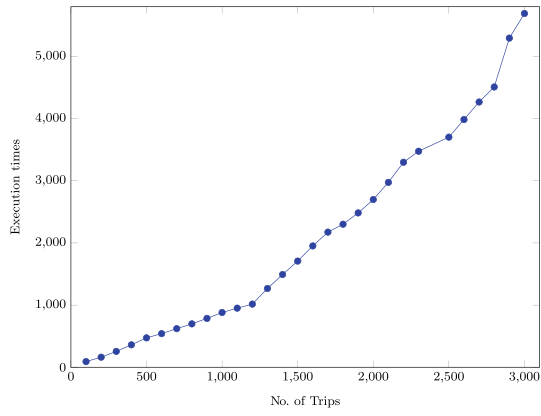


Fig. 8 Objective function values for modified numbers of depots

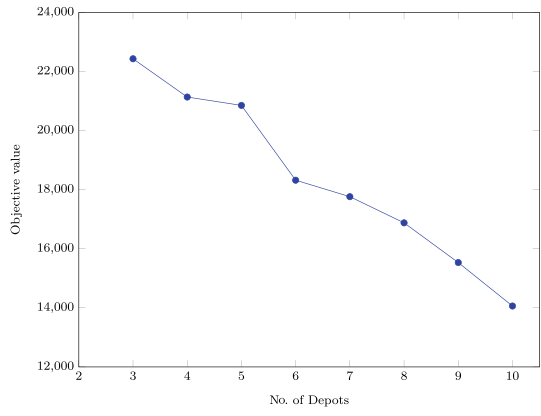
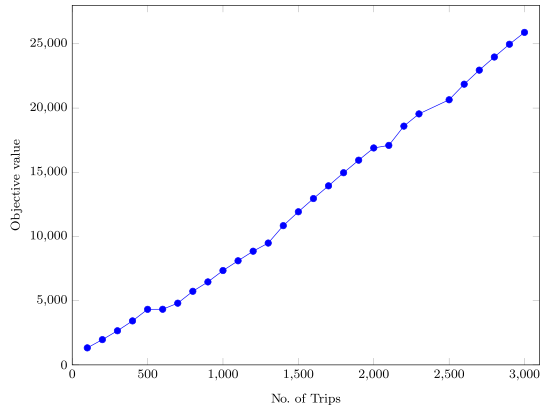


Fig. 9 Objective function values for modified numbers of trips



numbers of available vehicles at the depots v_d and the c_{ij}^d values are provided there for different numbers of depots (up to 10) as well as different numbers of trips (up to 3000). The remaining input is generated like before. Figures 6 and 7 show the change in the required execution time for the different numbers of depots and trips. The corresponding objective function values are shown in Figs. 8 and 9. Here, a considerable growth in the execution time by larger sets of depots and trips is observable.

The most important result, however, seems a positive answer to our research question R1 as we are able to solve related instances to optimality with a standard solver within time limits deemed practical rather than using the specialized (heuristic Benders) approach from Mesquita et al. (2013).

3.2 Extending to include buffer times

To focus on research question R2, we investigate the use of buffer times and delay propagation as presented in Amberg et al. (2019). Their model is flow-based. For each arc (i, j) , flow variables f_{ij}^d are considered indicating whether the arc (i, j) coming from depot d is used in the solution. Cost values c_{ij}^d with these variables are related to the (variable) costs of using a vehicle from depot d serving a specific arc (i, j) . If the model is assumed to be circular, fixed costs can be associated with the respective circulation arc; see Fig. 1. Delay propagation in Amberg et al. (2019) follows fixed buffer times or a calculated measure that represents the possible propagation of delays. Given a duty k with a set of trips to be performed, a measure r_k is defined incorporating expected “primary” delays and subsequent secondary delays ($k \in \mathcal{K}$ is supposed to perform a set of $T_k \subset \mathcal{T}$ of trips / tasks) as follows:

$$r_k = \sum_{i=1}^{|T_k|-1} p(t_i, t_{i+1})$$

$$p(t_i, t_{i+1}) = \max\{0, PAT(t_i) + PD(t_i) + p(t_{i-1}, t_i) - PDT(t_{i+1})\} \quad \forall k \in \mathcal{K} \tag{16}$$

where an index k is omitted where possible; $p(t_0, t_1)$ indicates the initial condition, i.e., the initial delay, say, of arriving at the start of a trip from a pull-out trip, $PAT(t_i)$ is the planned finishing time of trip or task t_i , $PD(t_i)$ is the expected primary delay of t_i and $PDT(t_{i+1})$ the planned departure time of the next trip t_{i+1} .

It should be noted that delay propagation of vehicles and/or crew duties is considered to be a measure of robustness (cf. Ge et al. 2020). In the spirit of general key performance indicators one may also ask to which extent robustness measures focusing on other stakeholders like passengers (see the above short discussion) may be incorporated into the model (beyond what is part of delay management).

Next, the integrated model of Mesquita et al. (2013) is enriched by adding some real elements coming from Amberg et al. (2019). The added elements provide the opportunity to embed buffer times between consecutive vehicle trips and delay propagation into the model.⁴

The formulation of this model is as follows:

$$\begin{aligned} \min \sum_{h \in H} \left(\sum_{d \in D} \sum_{(i,j) \in I^h} c_{ij}^{dh} z_{ij}^{dh} + \sum_{l \in L^h} e_l w_l^h \right) + \sum_{m \in M} \sum_{s \in S} r^m x_s^m + \lambda_T \eta_T + \lambda_O \eta_O \\ + \alpha_V \sum_{d \in D} \sum_{k \in K_V^d} ATD_k^d p_k^d + \alpha_C \sum_{d \in D} \sum_{k \in K_C^d} ATD_k^d q_k^d \end{aligned} \quad (17)$$

$$\sum_{d \in D} \sum_{i: (i,j) \in I^h} z_{ij}^{dh} = 1 \quad j \in N^h - D, h \in H \quad (18)$$

$$\sum_{i: (i,j) \in I^h} z_{ij}^{dh} - \sum_{i: (j,i) \in I^h} z_{ji}^{dh} = 0 \quad j \in N^h, d \in D, h \in H \quad (19)$$

$$\sum_{i \in N^h} z_{di}^{dh} \leq v_d \quad d \in D, h \in H \quad (20)$$

$$\sum_{l \in L_{ij}^h} w_l^h - \sum_{d \in D} z_{ij}^{dh} \geq 0 \quad (i,j) \in I^h, h \in H \quad (21)$$

$$\sum_{m \in M} y_l^{mh} - w_l^h = 0 \quad l \in L^h, h \in H \quad (22)$$

⁴ A direct use of the complete model of Amberg et al. (2019) would be a different case as different assumptions and legal constraints do not allow a direct comparison. Rather, we add from that model to gain insights on added richness towards the model of Mesquita et al. (2013). Adding days-off patterns as well as extended robustness issues towards the model of Amberg et al. (2019) is a topic for future research.

$$\sum_{s \in S} x_s^m \leq 1 \quad m \in M \tag{23}$$

$$\sum_{l \in L^h} y_l^{mh} - \sum_{s \in S} a_s^h x_s^m \leq 0 \quad m \in M, h \in H \tag{24}$$

$$\sum_{l \in L_f^h} y_l^{mh} + \sum_{l \in L_f^{h-1}} y_l^{m(h-1)} \leq 1 \quad m \in M, h \in H - \{1\}, f \neq g \in \{E, A\} \tag{25}$$

$$\sum_{h \in H} \sum_{l \in L_t^h} y_l^{mh} - \eta_t \leq 0 \quad m \in M, t \in \{T, O\} \tag{26}$$

$$\sum_{k \in K_{ij}^C} p_k^d - z_{ij}^{dh} = 0 \quad \forall d \in D, h \in H, \forall (i, j) \in I^h \tag{27}$$

$$\sum_{k \in K_{ij}^V} q_k^d - z_{ij}^{dh} = 0 \quad \forall d \in D, h \in H, \forall (i, j) \in I^h \tag{28}$$

$$DP_k = ED_k + \sum_{k'=1}^{k-1} p_{k'}^d \sigma_{k'}^d \quad d \in D, k \in K_{ij}^C \cup K_{ij}^V \tag{29}$$

$$ATD_k^d = \max(0, DP_k - BT_k) \quad d \in D, k \in K_{ij}^C \cup K_{ij}^V \tag{30}$$

$$z_{ij}^{dh} \in \{0, 1\} \quad (i, j) \in I^h, d \in D, h \in H \tag{31}$$

$$w_l^h \in \{0, 1\} \quad l \in L^h, m \in M, h \in H \tag{32}$$

$$w_l^h \in \{0, 1\} \quad l \in L^h, h \in H \tag{33}$$

$$y_l^{mh} \in \{0, 1\} \quad l \in L^h, m \in M, h \in H \tag{34}$$

$$x_s^m \in \{0, 1\} \quad s \in S, m \in M \tag{35}$$

$$\eta_T, \eta_O \geq 0 \text{ and integer} \tag{36}$$

$$p_k^d \in \{0, 1\} \quad \forall d \in D, \forall k \in K_V^d \tag{37}$$

Table 2 The generation rules of the input data

| Parameter | Generation rule |
|------------------------|--|
| c_{ij}^d | U[1,5] |
| e_i | U[5,15] |
| v_d | U[10,30] for small instances with enough depots; otherwise = $\frac{\text{the number of vehicles}}{\text{the number of depots}}$ |
| r^m | U[10,20] |
| a_s^h | = 1 if h is a workday on schedule s |
| λ_T, λ_O | = 2 |
| ED_k | U[1,15] |
| BT_k | U[1,5] |

$$q_k^d \in \{0, 1\} \quad \forall d \in D, \forall k \in K_C^d \quad (38)$$

The new objective function (17) measures the quality of the solution in terms of vehicle and driver costs, roster balancing, total buffer times, and delay propagation costs regarding the vehicle tasks and crew duties. Then constraints (2)–(10) are repeated as constraints (18)–(26) followed by the inclusion of the buffer times as well as the definition of the variable ranges.

Constraints (27) and (28) enforce that p_k^d and q_k^d must be equal to 1 for one k if $z_{ij}^{dh} = 1$. Constraint (29) sets the delay propagation until duty k as its expected delay plus the sum of all the actual delays of the previous duties. The actual delay of duty k as the maximum of zero and the propagated delay until then minus buffer time is calculated by constraint (30). The remaining constraints define the variable ranges.

To conduct numerical results, instances are generated as in the previous subsection with some necessary extensions regarding the problem modification. That is, the input data of the model is either randomly generated as shown in Table 2 or using the data provided in Huisman (2007) with the remaining data randomly generated. It is worth noting that we assume that the larger buffer times are corresponding to busier sections of the timetable.

Again, the model is programmed in GAMS with the CPLEX solver in standard mode on a computer as indicated above. For each instance size, ten instances are generated, five of them with totally random data and five based on the data from Huisman (2007). Whenever results are shown, the average of this sample is referred to.

Figures 10, 11, 12, and 13 depict the required execution times of the model with an increasing number of duties, vehicles, trips, and depots, respectively, once in its pure form and once by including fixed buffer times between the trips as well as delay propagation. In each case, the other parameters are set at their middle value. We set a time limit of 2 h or 7200 s for the solver. It means that whenever this limit is reached, the solution process stops. This time limit is shown in red in the figures.

As it is evident, adding the complementary elements to the model does not increase the execution time considerably. So, the model of Mesquita et al. (2013)

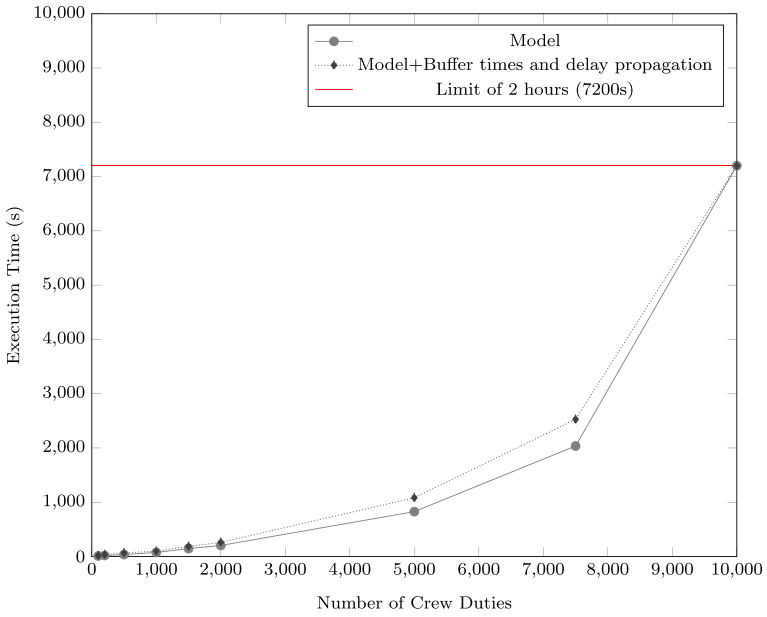


Fig. 10 The required execution time by increasing the number of duties

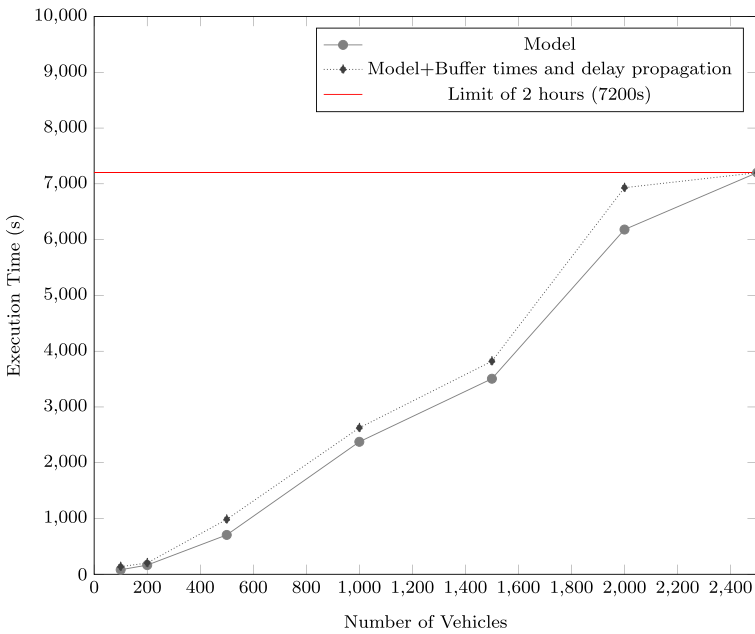


Fig. 11 The required execution time by increasing the number of vehicles

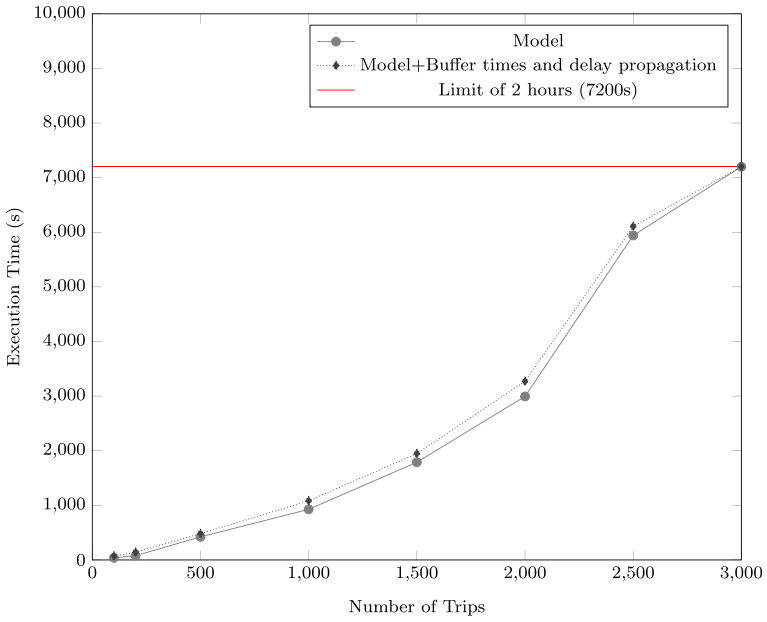


Fig. 12 Required execution times by increasing the number trips

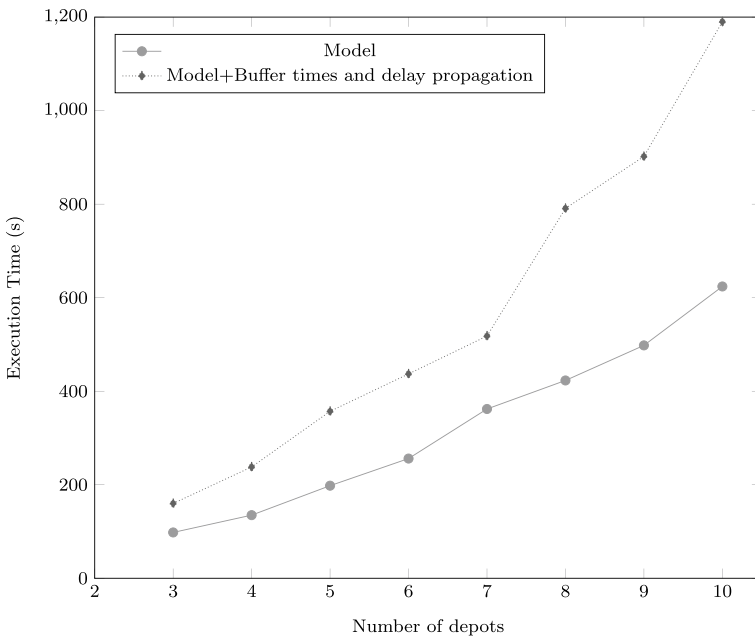


Fig. 13 Required execution times by increasing the number of depots

Fig. 14 Execution times by increasing the buffer times (*BT*)

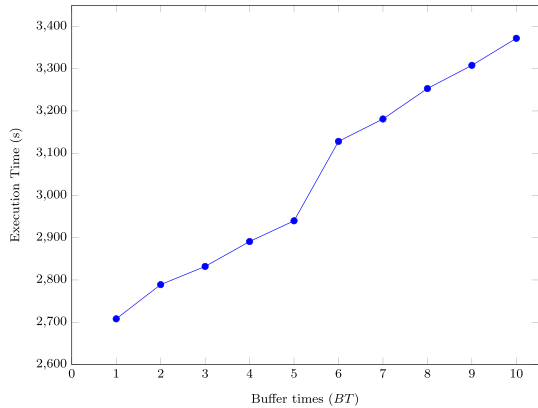
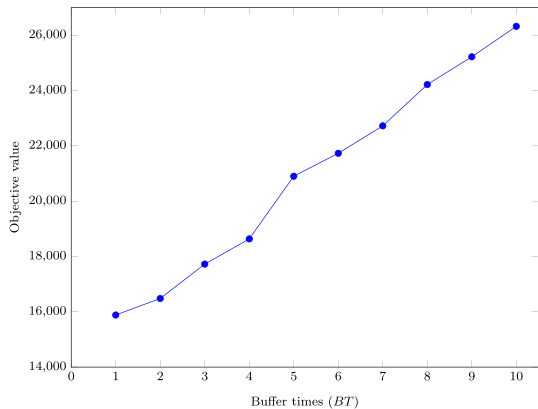


Fig. 15 Objective values by increasing the buffer times (*BT*)



can be easily enriched without being concerned about the solvability. By means of these experiments and the depicted results, we can positively answer the main question of this research (research question R2), which is about the solvability of this model by a standard exact solver. Although it is observed that the computation time grows considerably as any of the parameters increases, they can be at or near values known from practical settings. The interesting fact is the different trend observed in Fig. 13, which shows a substantial difference between the two models. This is due to the fact that the number of depots contributes to many constraint blocks. Therefore, increasing this parameter raises the total number of constraints in the model, which considerably slows down the process of exact optimization in the case that we are involved with the additional factor of robustness in the model.

At the end of this subsection, we report on some conducted experiments with increasing amounts of buffer times and the results in terms of required execution

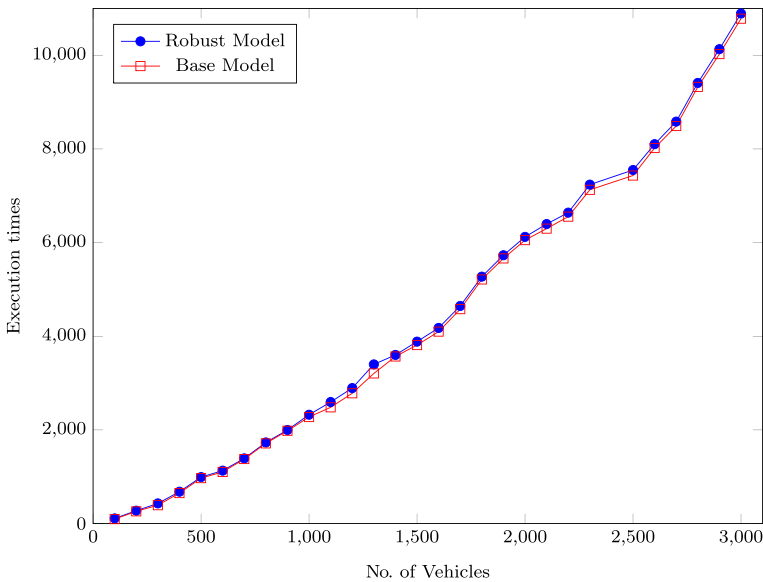


Fig. 16 Execution times (s) to achieve robust results vs. execution times of the base model for modified numbers of vehicles

times and objective values are shown in Figs. 14 and 15, respectively. As it can be observed, the objective function values worsen by ensuring longer buffer times and also the problem becomes harder. Therefore, more time is required to solve the problem.

3.3 Adding robustness to the model

Bearing in mind that the problem inputs frequently vary in the real world and have a non-deterministic nature, the concept of robustness is added to the model in this subsection. This is done by considering three different possible values or scenarios for all the coefficients existing in the objective function (1), which we call here Z_O . For this sake, each generated value of c_{ij}^d , e_l , r^m , λ_T and λ_O can be replaced by a set as $\{0.9 * fi, fi, 1.1 * fi\}$, where fi is the fixed input value. Then, numerous scenarios are generated by the combination of these possible coefficient values and the objective function value is calculated for each of them. The objective value of the robust model is the maximum or worst among the objective values of all scenarios because in this way we can ensure that in any case the objective cannot be worse than that. This is in accordance with the definition of robustness given in Ben-Tal et al. (2009). So, a robust version of the first model presented in Sect. 3.1 can be built by replacing the objective (1) or z with $\max_{All\ scenarios} z$. The same instances are solved for the robust models and the required execution times as well as the objective function values by a modified

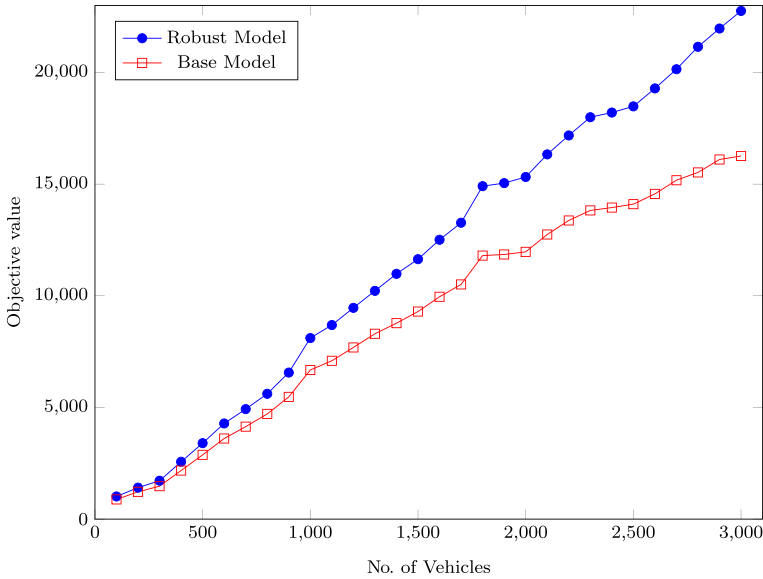


Fig. 17 Robust objective values vs. objective values of the base model for modified numbers of vehicles

number of vehicles are shown and compared with those of the base model in Figs. 16 and 17, respectively.

The robust results show that the execution time or the computational complexity of the problem does not considerably increase in comparison to the non-robust model. However, the objective value is moderately deteriorated in the robust model. This is expected because the worse values among so many cases are chosen (based on the different realizations of the variation with respect to the f_i -values).

4 Conclusions

In this research, the scalability of an extended integrated vehicle and crew scheduling (rostering) model from literature is examined. Extended problem settings incorporating buffer times and delay propagation do not harm the solvability of the model with standard solvers. The main achievement is the verification of the possibility of having new elements incorporated into a model that resorts to general purpose solvers rather than the need to use specially tailored algorithms. The results have been obtained for the same size of instances that have been tackled in the focused model from literature and beyond. A quote that we can borrow from Wolsey (2002) from the field of production planning and lot sizing from quite some time ago may even start holding in the realm of public transport: “there is a nontrivial fraction of practical [...] problems that can now be solved by nonspecialists just by taking an appropriate a priori reformulation of the problem, and

then feeding the resulting formulation into a commercial mixed-integer programming solver”.

As a direction for future research, we envisage the incorporation of stochastic elements with uncertainty sets regarding demand (beyond the first attempts of incorporating robustness as shown in this paper), the incorporation of load-dependent traffic situations as well as the consideration of an uncertain availability of vehicles. Besides, it would also be good to explore other ”richness” features such as rostering constraints involving a maximum number of days off within a month to discover their effect on the computation time, although it is expected that they increase the complexity of the problem non-trivially.

Another option for future work refers to setting up a general framework for replication studies. As this has not yet been extensively studied, we just propose to set it up along the lines provided in the literature review.

Acknowledgements We appreciate the comments and reports of the guest editors and the referees which helped to improve the quality of the paper.

Funding Open Access funding enabled and organized by Projekt DEAL. L. Ge appreciates the support from the German Research Foundation (DFG, fund LX 156/2-1).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abbink E, Fischetti M, Kroon L, Timmer G, Vromans M (2005) Reinventing crew scheduling at Netherlands railways. *Interfaces* 35(5):393–401. <https://doi.org/10.1287/inte.1050.0158>
- Amberg B, Amberg B, Kliwer N (2019) Robust efficiency in urban public transportation: Minimizing delay propagation in cost-efficient bus and driver schedules. *Transp Sci* 53:89–112. <https://doi.org/10.1287/trsc.2017.0757>
- Anand R, Aggarwal D, Kumar V (2017) A comparative analysis of optimization solvers. *J Stat Manag Syst* 20(4):623–635. <https://doi.org/10.1080/09720510.2017.1395182>
- Ang Pik Yoke J, Nourmohammadzadeh A, Shi X, Voß S (2021) Integrated vessel and crew scheduling in maritime shipping: Mathematical modelling and extended observations during the pandemic. Tech. rep. Institute of Information Systems (IWI), University of Hamburg
- Ben-Tal A, El-Ghaoui L, Nemirovsky A (2009) Robust optimization. Princeton University Press, Princeton
- Block J, Kuckertz A (2018) Seven principles of effective replication studies: strengthening the evidence base of management research. *Manag Rev Q* 68:355–359. <https://doi.org/10.1007/s11301-018-0149-3>
- Borndörfer R, Grötschel M, Jäger U (2010) Planning problems in public transit. In: Grötschel M, Lucas K, Mehrmann V (eds) *Production factor mathematics*. Springer, Berlin, pp 95–121. https://doi.org/10.1007/978-3-642-11248-5_6

- Borndörfer R, Löbel A, Reuther M, Schlechte T, Weider S (2013) Rapid branching. *Public Transp* 5:3–23. <https://doi.org/10.1007/s12469-013-0066-8>
- Boylan JE (2016) Reproducibility. *IMA J Manag Math* 27(2):107–108. <https://doi.org/10.1093/imaman/dpw003>
- Caserta M, Voß S (2021) Accelerating mathematical programming techniques with the corridor method. *Int J Prod Res* 59:2739–2771. <https://doi.org/10.1080/00207543.2020.1740343>
- Ceder A (2015) *Public transit planning and operation*, 2nd edn. CRC, Boca Raton
- Daduna J, Voß S (eds) (2000) *Informationsmanagement im Verkehr*. Physica, Heidelberg. <https://doi.org/10.1007/978-3-642-57682-9>
- de Armas J, Lalla-Ruiz E, Tilahun S, Voß S (2021) Similarity in metaheuristics: a gentle step towards a comparison methodology. *Natural Comput*. <https://doi.org/10.1007/s11047-020-09837-9>
- Desaulniers G, Hickman MD (2007) Public transit. In: Barnhart C, Laporte G (eds) *Transportation, handbooks in operations research and management science*, vol 14. Elsevier, pp 69–127. [https://doi.org/10.1016/S0927-0507\(06\)14002-5](https://doi.org/10.1016/S0927-0507(06)14002-5)
- Doi T, Nishi T, Voß S (2018) Two-level decomposition-based matheuristic for airline crew rostering problems with fair working time. *Eur J Oper Res* 267:428–438. <https://doi.org/10.1016/j.ejor.2017.11.046>
- Er-Rbib S, Desaulniers G, El Hallaoui I, Bani A (2021a) Integrated and sequential solution methods for the cyclic bus driver rostering problem. *J Oper Res Soc* 72:764–779. <https://doi.org/10.1080/01605682.2019.1700187>
- Er-Rbib S, Desaulniers G, El Hallaoui I, Munroe P (2021b) Preference-based and cyclic bus driver rostering problem with fixed days off. *Public Transp* 13:251–286. <https://doi.org/10.1007/s12469-021-00268-y>
- Fink A, Voß S (2002) HotFrame: a heuristic optimization framework. In: Voß S, Woodruff D (eds) *Optimization software class libraries*. Kluwer, Boston, pp 81–154. https://doi.org/10.1007/0-306-48126-X_4
- Fischetti M, Monaci M (2014) Exploiting erraticism in search. *Oper Res* 62(1):114–122. <https://doi.org/10.1287/opre.2013.1231>
- Fischetti M, Lodi A, Salvagnin D (2009) Just MIP it! In: Maniezzo V, Stützle T, Voß S (eds) *Matheuristics: hybridizing metaheuristics and mathematical programming*. Springer, Boston, pp 39–70. https://doi.org/10.1007/978-1-4419-1306-7_2
- Freling R (1997) *Models and techniques for integrating vehicle and crew scheduling*. PhD thesis, Tinbergen Institute, Erasmus University Rotterdam, Rotterdam
- Freling R, Wagelmans A, Paixão J (1999) An overview of models and techniques for integrating vehicle and crew scheduling. In: Wilson N (ed) *Computer-aided transit scheduling*. Springer, Berlin, pp 441–460. https://doi.org/10.1007/978-3-642-85970-0_21
- Freling R, Huisman D, Wagelmans A (2001) Applying an integrated approach to vehicle and crew scheduling in practice. In: Voß S, Daduna J (eds) *Computer-aided scheduling of public transport*. Springer, Berlin, pp 73–90. https://doi.org/10.1007/978-3-642-56423-9_5
- GAMS Development Corporation (2020) *General algebraic modeling system (GAMS) Release 32.2.0*. <http://www.gams.com/>
- Ge L, Voß S, Xie L (2020) Robustness and disturbances in public transport. Tech. rep., Institute of Information Systems, Leuphana University of Lüneburg and Institute of Information Systems (IWI), University of Hamburg
- Ge L, Sarhani M, Voß S, Xie L (2021) Review of transit data sources: potentials, challenges and complementarity. *Sustainability* 13(20):11450. <https://doi.org/10.3390/su132011450>
- Gleixner A, Hendel G, Gamrath G, Achterberg T, Bastubbe M, Berthold T, Christophel P, Jarck K, Koch T, Linderoth J, Lübbecke M, Mittelmann HD, Ozyurt D, Ralphs TK, Salvagnin D, Shinano Y (2021) *MIPLIB 2017: data-driven compilation of the 6th mixed-integer programming library*. *Math Program Comput*. <https://doi.org/10.1007/s12532-020-00194-3>
- Haase K, Desaulniers G, Desrosiers J (2001) Simultaneous vehicle and crew scheduling in urban mass transit systems. *Transp Sci* 35:286–303. <https://doi.org/10.1287/trsc.35.3.286.10153>
- Hartl RF, Hasle G, Janssens GK (2006) Special issue on rich vehicle routing problems. *Cent Eur J Oper Res* 14(2):103–104. <https://doi.org/10.1007/s10100-006-0162-9>
- Himmich I, El Hallaoui I, Soumis F (2020) Primal column generation framework for vehicle and crew scheduling problems. *Networks* 75(3):291–309. <https://doi.org/10.1002/net.21925>
- Hüffmeier J, Mazei J, Schultze T (2016) Reconceptualizing replication as a sequence of different studies: a replication typology. *J Exp Soc Psychol* 66:81–92. <https://doi.org/10.1016/j.jesp.2015.09.009>

- Huisman D (2007) <https://personal.eur.nl/huisman/instances.htm>. 26 Dec 2020
- Huisman D, Freling R, Wagelmans APM (2005) Multiple-depot integrated vehicle and crew scheduling. *Transp Sci* 39:491–502. <https://doi.org/10.1287/trsc.1040.0104>
- Jörges S, Steffen B (2014) Back-to-back testing of model-based code generators. *Lect Notes Comput Sci* 8802:425–444. https://doi.org/10.1007/978-3-662-45234-9_30
- Kendall G, Bai R, Błazewicz J, De Causmaecker P, Gendreau M, John R, Li J, McCollum B, Pesch E, Qu R, Sabar N, Vanden Berghhe G, Yee A (2016) Good laboratory practice for optimization research. *J Oper Res Soc* 67:676–689. <https://doi.org/10.1057/jors.2015.77>
- Koch T, Achterberg T, Andersen E, Bastert O, Berthold T, Bixby R, Danna E, Gamrath G, Gleixner AM, Heinz S, Lodi A, Mittelmann H, Ralphs T, Salvagnin D, Steffy D, Wolter K (2011) MIPLIB 2010. *Math Program Comput* 3, paper 103. <https://doi.org/10.1007/s12532-011-0025-9>
- Lalla-Ruiz E, Voß S (2016) Improving solver performance through redundancy. *J Syst Sci Syst Eng* 25:303–325. <https://doi.org/10.1007/s11518-016-5301-9>
- Lam E, Van Hentenryck P, Kilby P (2020) Joint vehicle and crew routing and scheduling. *Transp Sci* 54(2):488–511. <https://doi.org/10.1287/trsc.2019.0907>
- López-Ibáñez M, Branke J, Paquete L (2021) Reproducibility in evolutionary computation. *Computing Research Repository (CoRR)* <https://arxiv.org/abs/2102.03380>
- Mesquita M, Moz M, Paías A, Paixão J, Pato M, Respício A (2011) A new model for the integrated vehicle-crew-rostering problem and a computational study on rosters. *J Sched* 14(4):319–334. <https://doi.org/10.1007/s10951-010-0195-8>
- Mesquita M, Moz M, Paías A, Pato M (2013) A decomposition approach for the integrated vehicle-crew-roster problem with days-off pattern. *Eur J Oper Res* 229:318–331. <https://doi.org/10.1016/j.ejor.2013.02.055>
- Mittelmann H (2020) Benchmarking optimization software—a (hi)story. *SN Operations Research Forum* 1:paper 2. <https://doi.org/10.1007/s43069-020-0002-0>
- Nissen V (2018) Digital transformation of the consulting industry—introduction and overview. In: Nissen V (ed) *Digital transformation of the consulting industry*. Springer, Cham, pp 1–58. https://doi.org/10.1007/978-3-319-70491-3_1
- Otto A, Agatz N, Campbell J, Golden B, Pesch E (2018) Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: a survey. *Networks* 72(4):411–458. <https://doi.org/10.1002/net.21818>
- Perumal S, Lusby R, Larsen J (2020) A review of integrated approaches for optimizing electric vehicle and crew schedules. *Tech. rep.*, DTU, Lyngby
- Perumal S, Larsen J, Lusby R, Riis M, Christensen T (2021) A column generation approach for the driver scheduling problem with staff cars. *Public Transp*. <https://doi.org/10.1007/s12469-021-00279-9>
- Shen Y, Xia J (2009) Integrated bus transit scheduling for the Beijing bus group based on a unified mode of operation. *Int Trans Oper Res* 16(2):227–242. <https://doi.org/10.1111/j.1475-3995.2009.00673.x>
- Sörensen K, Arnold F, Palhazi Cuervo D (2019) A critical analysis of the “improved Clarke and Wright savings algorithm”. *Int Trans Oper Res* 26(1):54–63. <https://doi.org/10.1111/itor.12443>
- Swan J, Adriaensens S, Brownlee A, Hammond K, Johnson C, Kheiri A, Krawiec F, Merelo J, Minku L, Özcan E, Pappa G, García-Sánchez P, Sörensen K, Voß S, Wagner M, White D (2022) Metaheuristics “in the large”. *Eur J Oper Res* 297:393–406. <https://doi.org/10.1016/j.ejor.2021.05.042>
- Swierstra A, van Nes R, Molin E (2017) Modelling travel time reliability in public transport route choice behaviour. *Eur J Transp Infrastruct Res* 17(2):263–278. <https://doi.org/10.18757/ejtir.2017.17.2.3194>
- Tahir A, Desaulniers G, El Hallaoui I (2019) Integral column generation for the set partitioning problem. *Eur J Transp Logist* 8:713–744. <https://doi.org/10.1007/s13676-019-00145-6>
- Taylor SJE, Eldabi T, Monks T, Rabe M, Uhrmacher AM (2018) Crisis, what crisis—does reproducibility in modeling and simulation really matter? In: 2018 winter simulation conference (WSC), pp 749–762. <https://doi.org/10.1109/WSC.2018.8632232>
- Voß S, Lalla-Ruiz E (2016) A set partitioning reformulation for the multiple-choice multidimensional knapsack problem. *Eng Optim* 48(5):831–850. <https://doi.org/10.1080/0305215X.2015.1062094>
- Vuchic VR (2005) *Urban transit: operations, planning, and economics*. Wiley, Hoboken
- Wessel N, Widener M (2017) Discovering the space-time dimensions of schedule padding and delay from GTFS and real-time transit data. *J Geogr Syst* 19:93–107. <https://doi.org/10.1007/s10109-016-0244-8>
- Wolsey LA (2002) Solving multi-item lot-sizing problems with an MIP solver using classification and reformulation. *Manag Sci* 48(12):1587–1602. <https://doi.org/10.1287/mnsc.48.12.1587.442>

Xie L, Suhl L (2015) Cyclic and non-cyclic crew rostering problems in public bus transit. *OR Spectrum* 37:99–136. <https://doi.org/10.1007/s00291-014-0364-9>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.