



A shift scheduling model for ridepooling services

Lukas Berthold¹ · Malte Fliedner¹  · Arne Schulz^{1,2}

Received: 25 September 2023 / Accepted: 28 July 2024
© The Author(s) 2024

Abstract

The planning of efficient shift schedules is a key challenge for many service companies whose economic success heavily relies on the efficient employment of personnel. In spite of the recent advances in autonomous driving, mobility services, such as ride pooling, still heavily rely on the use of human drivers and will presumably remain in this category in the near to midterm. As a consequence, shift scheduling of drivers is one of the key success factors in the current industry environment. Determining appropriate shifts that minimize an under- and oversupply of vehicles for all planning periods is a challenging task, since demand can vary heavily over time and the assignment flexibilities are limited due to driver preferences and regulations. In this work, we present a shift scheduling model for ridepooling services. Moreover, we introduce a data generator for instances with realistic properties of a ridepooling service. Using it, we study the effect of different kinds of flexibilities on solution quality.

Keywords Shift planning · Ridepooling · On-demand transportation · Mixed-integer programming

1 Introduction

Ridepooling is an emerging market in the mobility sector, especially as it allows to address three dimensions of sustainability. Ridepooling providers usually operate with emission-free electric vehicles (ecological responsibility). Moreover, their on-demand character can allow for an efficient implementation (economic

✉ Malte Fliedner
malte.fliedner@uni-hamburg.de

Arne Schulz
arne.schulz@uni-hamburg.de; arne.schulz@hsu-hh.de

¹ Institute of Operations Management, Universität Hamburg, Moorweidenstraße 18, 20148 Hamburg, Germany

² Institute of Quantitative Logistics, Helmut Schmidt University, Holstenhofweg 85, 22043 Hamburg, Germany

performance) in regions which are not well-connected to the public transport network such that mobility becomes more available to all persons (social equity). Ridepooling services typically employ small vans with up to six seats. A customer books its ride using an app, where the user also sees the starting time and an arrival interval (arrival times may vary if the trip is combined with other customers later on) for the ride. Due to the small vehicles and the flexible character of the service, ridepooling services allow to address all three mentioned dimensions of sustainability if the service is well-planned. In the current industry environment, there are two main planning problems that service providers need to master. In the short term, incoming customer requests need to be efficiently assigned to vehicles to increase the utilization of shared resources. This problem is known as the dial-a-ride problem (see Ho et al. (2018) for a survey). Methods to solve the problem for the application of a ridepooling service can be found in Pfeiffer and Schulz (2022a) (heuristic) and Schulz and Pfeiffer (2024) (exact). Before the vehicle scheduling can take place however, the service provider needs to make sure that a sufficient number of vehicles and drivers are in the field to service customer demand.

Although ridepooling services plan to operate with autonomous vehicles in the long term, it is currently unclear when autonomous vehicles will be used on public roads in large quantities. Hence, ridepooling providers face the challenging task to schedule divers' shifts covering the demand while being attractive for drivers. There are three main reasons making it difficult to schedule attractive shifts while fulfilling the demand: First, larger providers work with several hundreds of drivers leading to instances of significant size. Second, demand is highly fluctuating (see (Kuehnel et al. 2021, especially Figure 6) for a typical demand curve), as ridepooling services are particularly used in rush hours and leisure times in the evening or at night (Kostorz et al. 2021). Third, service times vary over the week. As an example some providers like HolMichApp in the middle-sized German city Wuppertal vary their service times by a later start on Sundays (6am – 10pm on every day but on Sundays from 8am on) as well as an extended service on Fridays and Saturdays (until 3am).¹ Other providers like hvp hop in the small city Ahrensburg, which is close to Hamburg in Germany, vary their starting times stronger (4:30am on Mondays to Fridays, 5:15am on Saturdays, and 8am on Sundays) and also have different closing hours (0:30am on Mondays to Saturdays and 11pm on Sundays).² Although different starting and closing hours lead already to difficulties for shift scheduling, it becomes even more challenging if the service operates on some days around-the-clock but is closed over night on other days. An example is MOIA which operates amongst others in Hamburg. Their service operates on Mondays to Wednesdays from 5am to 1am as well as from Thursdays 5am until next Sunday 6am and from Sundays 8am until midnight.³ It is intuitive that it is challenging to schedule regular shifts of five, seven or eight hours without a too large variation in working times over the week such that demand in the nights between Thursday and Sunday can be covered while

¹ <https://www.holmich-app.de/>, last access on July 21st 2023 (in German)

² <https://vhhbus.de/hop/on-demand-shuttle-kreis-stormarn/>, last access on July 21st 2023 (in German)

³ <https://www.moia.io/en/cities>, last access on July 21st 2023.

the service does not operate on the other weekdays' nights—especially as there is a high demand in late evening and night hours on weekends (Kostorz et al. 2021, Figure 5).

To the best of our knowledge this paper describes for the first time a shift scheduling model solving this challenging task of ridepooling providers. Kuehnel et al. (2021) found out that in a comparison with a setting with autonomous vehicles 26% less rides could be served with working shifts. This indicates that an optimal shift planning is crucial for a ridepooling service because the difference between manually driven vehicles and autonomous vehicles is that autonomous vehicles can be used completely flexible.

Beside the introduction of a basic mixed-integer programming model for the shift scheduling of a ridepooling service, we develop two extensions to the base model, in order to implement different degrees of flexibility in the shift's starting times over the week and to include additional shifts from a working-time account, to face the aforementioned challenges. Furthermore, we use the demand curve representing the demand of the ridepooling service MOIA as presented in Kostorz et al. (2021) to develop a data generator for the demand on vehicles in a ridepooling service. This generator allows us and other researchers to evaluate shift scheduling models for ridepooling services facing all three challenges mentioned above on realistic instances. We use it to evaluate our model and its extensions.

The paper is constructed as follows: In Sect. 2, we review the relevant literature before the problem setting is described in detail in Sect. 3. Afterwards, we present our mixed-integer programming solution approach for the shift scheduling problem as well as the aforementioned extensions (Sect. 4). In Sect. 5, we introduce the data generator and conduct a computational study to analyze the model performance and the effect of the introduced kinds of flexibility on the solution quality. The paper closes with a conclusion in Sect. 6.

2 Literature review

Personnel scheduling has been one of the central applications of operations research methods for many decades now and there is a plethora of proposals for a vast amount of differing problem settings (e.g. see the review of Van den Bergh et al. (2013)). Planning models are developed for long term workforce planning which typically considers time-spans of up to a single or several years as well as for short term planning of daily shifts and their assignment to the workforce. With respect to long term planning, Azmat and Widmer (2004), for instance, developed a shift scheduling model for workforce planning on the annual level. They assumed that workers can perform all activities in their sector and that demand has to be satisfied. Moreover, they included holiday weeks. Hertz et al. (2010) considered annualized working hours in their workforce planning model using flexibility to respond to demand fluctuations. Moreover, they allowed gradual hiring of workers.

Short term planning approaches rather focus on an appropriate scheduling of existing employees and often seek to consider the preferences of employees in the decision making. They are often further subdivided into single-day shift scheduling

(or time of day scheduling), which seeks to find an acceptable combination of shift patterns for a single working day, days-off scheduling (also day-of-week scheduling), which seeks to cover an operating week of a production or service facility by several overlapping working weeks of employees, and tour scheduling, which constitutes a combination of the two aforementioned problems and seeks to find an appropriate set of consecutive daily shifts and days-off for each employee (Van den Bergh et al. (2013)). This last class of problems is considered the most complex of the personnel scheduling models and is the subject of this paper. Since the terminology employed in the literature somewhat varies (other terms sometimes employed for tour scheduling are for instance staff scheduling or rostering), we will simply use the term shift scheduling to refer to all of these approaches.

The exact structure of planning models can differ significantly with respect to the field of application. A plurality of shift scheduling approaches is devoted to hospitals to schedule physicians or nurses. Typical in this environment is that the service is never interrupted by planned downtime, shift lengths and patterns are often standardized and heterogeneous skills of employees need to be taken into account (see the surveys of Erhard et al. (2018) and Burke et al. (2004)). Further important factors that are addressed in planning approaches is the minimization of overtime (e.g., see Brunner et al. (2009)) or the inherent trade-off between service quality and working conditions. El-Rifai et al. (2015), for instance, balance this trade-off by optimizing the shift distribution among personnel and minimizing patients' expected waiting time.

From the methodical point of view, mixed-integer programming is an often used method in shift scheduling. Amongst other topics it is applied to problems from the area of transportation planning like our shift scheduling problem for drivers of a ridepooling service. Carotenuto et al. (2019) developed a mixed-integer program for employee scheduling in aircraft refueling. Cavada et al. (2020) used mixed-integer programming for workforce planning in the outbound baggage loading area of an international airport. Moreover, Horn et al. (2021) used a mixed-integer program for a daily driver shift scheduling problem for an online store. A review on staff scheduling and rostering in transportation systems can be found in Ernst et al. (2004).

Ladier et al. (2014) solved the weekly timetabling and daily rostering problem of a logistics company with a three step mixed-integer programming model. First, the workforce is dimensioned. Second, tasks are allocated on a weekly level and third, the detailed daily rostering is determined. An integrated problem in logistics was also investigated by Restrepo et al. (2019). They engaged in an integrated shift scheduling and load assignment problem with an application in attended home delivery. The relation between customer demand and shift planning was included by Kabak et al. (2008). For a setting in the retail sector they first determined staff demand in the store and then optimized the shift scheduling.

A large number of shift scheduling approaches optimizes staffing cost, which makes sense whenever staffing levels are part of the decision problem. However, once staffing levels are fixed, in particular in service systems, it might make more sense to focus on an ideal demand coverage which avoids undersupply and oversupply of staff in the planning periods. Several approaches sought to consider

worker flexibility in order to avoid over-/undersupply. For instance, Topaloglu and Ozkarahan (2004) developed an implicit goal programming approach for shift scheduling considering flexible start times, shift lengths, breaks, and days-off as well as worker preferences in the context of nurse scheduling. Henao et al. (2010) investigated the inherent flexibility provided by multi-skilled workers on shift scheduling in the retail sector. More recently, Álvarez et al. (2020) optimized over- and undersupply of staff in retailing and made use of the flexibility of scheduling multiple breaks per shift to achieve this goal.

Beside the mentioned flexibilities of starting times and breaks, working-time accounts with additional shifts and additional days-off are a further option to react on fluctuating demand. Pandey et al. (2021) considered days-off scheduling and used deterministic finite automata to generate a set of promising shift profiles for later optimisation. A linear programming model comprising working-time accounts was introduced by Corominas et al. (2010). Sillekens et al. (2011) used a linear approximation to represent the working-time account. Corominas et al. (2012) developed an aggregated planning model including hiring, firing, overtime, and working-time accounts. Pastor and Olivella (2008) investigated the case of a retail clothing chain. In their working-time accounts, a certain limit cannot be exceeded. Otherwise, the company has to pay for all positive hours in the working-time account such that it is set back to 0.

Compared with the vast majority of approaches considered in the literature, the highly fluctuating demand in ridepooling services makes very small planning intervals of 15 min necessary. In contrast to that, the planning horizon is quite long and lasts up to 4 weeks. The demand pattern has an observable and repeatable weekly structure (see Sect. 5.1), so that in principle cyclic approaches, such as the one presented in Aykin (2000), might be suitable to capture a large portion of the demand structure. Yet, the planning needs to also consider expected deviations from the general demand pattern (e.g. due to weather influences or special events). Furthermore, there are additional organisational restrictions which need to be observed. We thus set out to develop a new base model for this problem setting and additionally consider the flexibility of starting times as well as the option of additional shifts. Even though we will discuss shift planning mainly in the context of a ride pooling service operator, the model is sufficiently general to be of interest for ride hailing services as well, as long as the drivers are employed at the service operator and the operator seeks to optimize service coverage over a given planning horizon. A detailed description of our problem setting is presented in the following section.

3 Problem description

We consider a given time horizon $|T|$ distinguished in equal-sized periods $t = 1, \dots, |T|$ of 15 min length and a given number of drivers. Our model periodically repeats a weekly schedule. Thus, we assume that the time horizon $|T|$ is a multiple of a week. Each driver has a pattern describing its general working structure. The pattern indicates the number of hours \hat{h} the driver works in a shift and the number of days \hat{d} the driver

works consecutively per week. Thus, the driver's pattern is described by $s = (\hat{h}, \hat{d})$. The set of all shift patterns is given by S . Shifts are connected and only interrupted by a 30 min break after half of the shift if the shift is at least six hours long. In line with the practice, we assume that no further stops are necessary during the shift to refuel or charge the vehicle.

The demand d_t is the demand of drivers/vehicles in period t , which we assume to be given. This means that we assume a further planning step to be done beforehand to determine the demand curve for the required vehicles. In this planning step, the provider uses the expected demand of rides to determine the number of required vehicles per time period. Note that our demand curve for the required vehicles depends on the solution of this challenging previous planning step distributing the expected customer demand to vehicles and thereby decide on the routing of the vehicles. The demand curve for the required vehicles also depends on the investigated setting. In the case of ridepooling, rides can be shared such that the number of required vehicles is reduced in comparison to a taxi setting where no sharing is allowed (Pfeiffer and Schulz (2022b)).

The ridepooling provider operates with depots $b \in B$. The capacity of each depot is bounded by cap_b , i.e. at most cap_b drivers are assigned to depot $b \in B$. Moreover, we restrict the number of shifts starting in each time period to t^{max} . This is an organizational constraint, as starting drivers are concentrated around the depots and need some time to be distributed over the service area to be deployed optimally.

The objective is to schedule the shifts' starting times such that the demand of vehicles is approximated as good as possible given the drivers of different shift types over all time periods. Thus, we minimize the deviation of the demand curve, which expresses the number of required vehicles in every period, by excess supply as well as undersupply. Note that a perfect approximation of the vehicles' demand curve does not mean that all customer requests can be served, as the vehicles' demand curve is an approximation determined given the expected customer demand.

In the basic model (Sect. 4.1), we require that each driver starts all its shifts at the same time of the day. Later, we allow a restricted variation of the starting times of a driver's shifts during the week (Sect. 4.2). Moreover, we extend the model to integrate additional shifts (Sect. 4.3). Thus, the weekly schedule of a driver can vary over different weeks. Although we do not assign drivers directly to shifts, we outline in Sect. 4.5 how shifts can be distributed fairly amongst drivers.

The next section introduces our mixed-integer programming model.

4 Mixed-integer programming model

The following table summarizes the notation used in the mixed-integer program (Table 1).

If a shift is at least six hours long, we schedule a break of 30 min after half of the shift's duration, i.e. with $q = T^d/48$

Table 1 Notation

Indices	
$b \in B$	Depot
d	Day
$s \in S$	$s = (\hat{h}, \hat{d})$ shift pattern with \hat{h} working hours on \hat{d} consecutive days
$t \in T$	time periods with $T = \{1, \dots, T \}$
Parameters and scalars	
cap_b	Maximal number of tours assigned to depot b
c_t^o	Weight for excess supply in period t
c_t^u	Weight for undersupply in period t
d_t	Demand in period t
h^s	Maximal number of cancelled or added shifts per month and driver (used to model additional shifts)
$I_{tt'}^s$	Is 1 if a driver with shift pattern s starting in period t is active in period t'
q_s	Maximal number of tours, i.e. number of drivers, with shift pattern s
t^{max}	Maximal number of shifts starting in one period $t \in T$
T^d	Number of periods per day
T^w	Number of periods per week
v	Permitted deviation of starting time (one direction)
Integer variables	
a_{st}	Number of tours of shift pattern s scheduled to start in period t
f_{st}^-	Number of shifts of shift pattern s starting in period t with an extra day before the tour starts (used to model additional shifts)
f_{st}^+	Number of shifts of shift pattern s starting in period t with an extra day after the tour ends (used to model additional shifts)
g_{st}^b	Number of tours of shift pattern s starting in period t which are assigned to depot b
x_{st}	Number of shifts of shift pattern s active in period t
y_{st}	Number of shifts of shift pattern s starting in period t
Continuous variables	
o_t	Excess capacity in period t
u_t	Undersupply in period t

$$I_{tt'}^s = \begin{cases} 1, & \text{if } t' \in [t, t + q \cdot \hat{h}] \cup [t + q \cdot \hat{h} + q, t + 2q \cdot \hat{h} + q] \\ & \text{and } s = (\hat{h}, \hat{d}) : \hat{h} \geq 6, t \in \{1, \dots, |T| - 2q \cdot \hat{h} - q\}, \\ 1, & \text{if } t' \in [t, t + 2q \cdot \hat{h}] \text{ and } s = (\hat{h}, \hat{d}) : \hat{h} < 6, t \in \{1, \dots, |T| - 2q \cdot \hat{h}\}, \\ 0, & \text{else.} \end{cases}$$

As $2q = T^d/24$ is the number of periods per hour and \hat{h} the shift's duration, time interval $[t, t + 2q \cdot \hat{h}]$ starts in t and ends \hat{h} hours later at the end of the shift. Analogously, $q = T^d/48$ equals the number of periods in 30 min. Thus, time interval $[t, t + q \cdot \hat{h}]$ comprises the first half of the shift while $[t + q \cdot \hat{h} + q, t + 2q \cdot \hat{h} + q]$ starts half an hour after the first interval ended and ends at the shifts end, i.e. \hat{h} hours plus q periods later than the shift's start.

4.1 Basic model

With the above notation the model formulation is as follows:

$$\min \sum_{t \in T} c_t^o \cdot o_t + c_t^u \cdot u_t \quad (1)$$

with the constraints

$$\sum_{s \in S} x_{st} - d_t \leq o_t \quad \forall t \in T \quad (2)$$

$$d_t - \sum_{s \in S} x_{st} \leq u_t \quad \forall t \in T \quad (3)$$

$$\sum_{s \in S} x_{st} = 0 \quad \forall t \in T : d_t = 0 \quad (4)$$

$$x_{st} = \sum_{t': I_{t'}^s = 1} y_{st'} \quad \forall t \in T, s \in S \quad (5)$$

$$y_{st} = \sum_{d=1}^{\hat{d}} a_{s,t-(d-1) \cdot T^d} \quad \forall t \in T, s \in S \quad (6)$$

$$\sum_{t=1}^{T^w} a_{st} \leq q_s \quad \forall s \in S \quad (7)$$

$$a_{st} = a_{s,t+T^w} \quad \forall t = 1, \dots, |T| - T^w, s \in S \quad (8)$$

$$\sum_{b \in B} g_{st}^b = a_{st} \quad \forall t = \{1, \dots, T^w\}, s \in S \quad (9)$$

$$\sum_{s \in S} \sum_{t=1}^{T^w} g_{st}^b \leq \text{cap}_b \quad \forall b \in B \quad (10)$$

$$\sum_{s \in S} y_{st} \leq t^{\max} \quad \forall t \in T \quad (11)$$

$$o_t, u_t \geq 0 \quad \forall t \in T \quad (12)$$

$$g_{st}^b \in \mathbb{N}_0 \quad \forall t \in T, s \in S \tag{13}$$

$$a_{st} \in \mathbb{N}_0 \quad \forall t \in T, s \in S \tag{14}$$

Objective function (1) minimizes the deviation between supply and demand. Thereby, the deviation can be weighted individually for over- and undersupply as well as for each period by choosing parameters c_t^o and c_t^u . Over- and undersupply are determined by (2) and (3) with the total supply $\sum_{s \in S} x_{st}$ in period $t \in T$. Periods with a demand of 0 are considered as a special case because a demand of 0 means that we are in a closing hour. Thus, the supply has to be 0 as well (compare (4)). Constraints (5) determine the supply by summing up all active tours. A tour is active if the shift does not start later than t , does not end before t , and has no break in t (compare choice of I_{tt}^s). Besides, a tour is active if it started on an earlier day, was active in the time period representing the same time on that day, and has no day-off on the day of period t (compare (6)). Due to Constraints (7) the number of started tours over all time periods is restricted by the total number of drivers employed with the corresponding shift pattern. We only need to consider the first T^w periods here, as every driver has a positive working time in a week and thus needs to be scheduled within the first week. The remaining weeks of the time horizon are reflections of the first week, i.e. all drivers have the same schedule in each week. Constraints (8) ensure this point. Constraints (9) and (10) view the assignment of drivers to depots. While Constraints (9) take care that all shifts/drivers are assigned to a depot, Constraints (10) guarantee that the depot's capacity is held. Constraints (11) are an organizational restriction. It restricts the number of shifts starting in a single time period. This is a useful restriction, as starting drivers are concentrated in the neighbourhood of the depots directly after their shift's start. Thus, we restrict the number of simultaneously starting shifts to ensure that not a large amount of drivers cannot be deployed to satisfy customer demand because they are clustered. Finally, Constraints (12)–(14) are the non-negativity and integer constraints. Note that there is an upper bound on a_{st} and g_{st}^b by Constraints (7) and Constraints (9)–(10) and x_{st} as well as y_{st} are implicitly integer due to Constraints (5) and Constraints (6).

4.2 Flexible starting times

The basic model only allows fixed starting times. A single driver starts all its shifts at the same time of the day. Although having the same starting time at each day is beneficial for drivers, it makes the schedule inflexible. Consider the demand at a certain point in time of the week over all weeks of the planning horizon. If the demand at this time of the week varies over the different weeks, we cannot meet the demand exactly in every week. To face this issue we allow slightly varying starting times in this section.

Figure 1 gives an example for a driver's week schedule with fixed and varying starting times. While the driver starts its shift every day at 9am on the left side, starting times vary in a time interval of two hours around 9am on the right side.

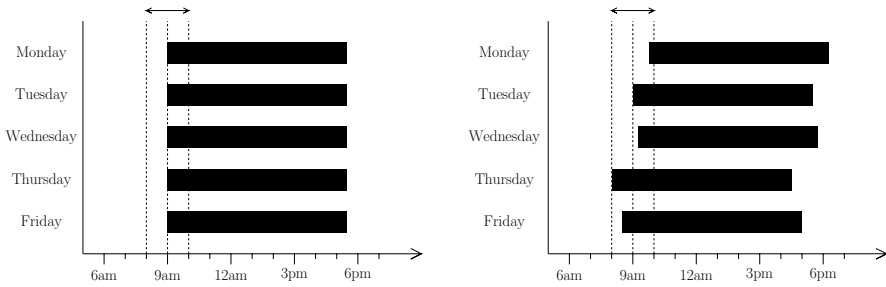


Fig. 1 Example for a week schedule of a driver with fixed (left) and varying (right) starting times

We set the allowed starting time deviation in one direction (in the figure one hour) around the scheduled starting time as parameter v . Thus, manager and drivers can reach an agreement on the extent of starting time deviations.

To integrate starting time deviations into the model, we replace (6) by

$$\sum_{t'=1}^t y_{st'} \geq \sum_{t'=1}^{t-v} \sum_{d=1}^{\hat{d}} \mathbf{1}_{\{(d-1) \cdot T^d + t' \leq t-v\}} a_{st'} \quad \forall t \in T, s \in S \tag{15}$$

$$\sum_{t'=1}^t y_{st'} \leq \sum_{t'=1}^{t+v} \sum_{d=1}^{\hat{d}} \mathbf{1}_{\{(d-1) \cdot T^d + t' \leq t+v\}} a_{st'} \quad \forall t \in T, s \in S \tag{16}$$

$$y_{st} \in \mathbb{N}_0 \quad \forall t \in T, s \in S \tag{17}$$

where $\mathbf{1}_{\{\cdot\}}$ is the indicator function, i.e.

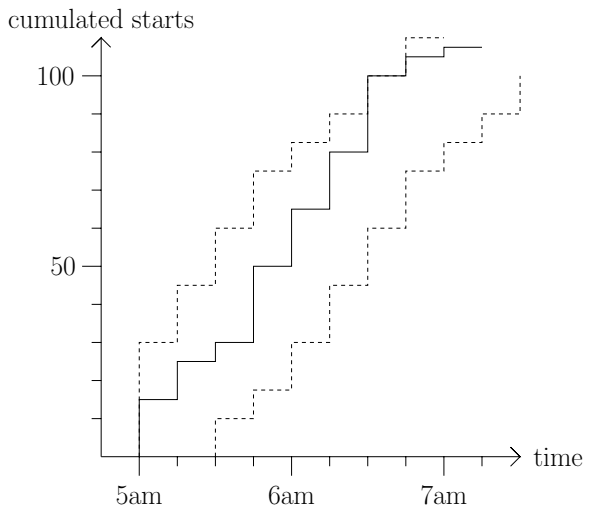
$$\mathbf{1}_{\{(d-1) \cdot T^d + t' \leq t-v\}} = \begin{cases} 1, & \text{if } (d-1) \cdot T^d + t' \leq t-v, \\ 0, & \text{else.} \end{cases}$$

In the basic model, all tours which are scheduled to start in period t start in period t (a_{st}). Now, Constraints (15) and (16) restrict the number of started shifts of shift pattern s in both directions like the following reformulation (for any $t \in T$ and $s \in S$) shows

$$\sum_{t'=1}^{t-v} \sum_{d=1}^{\hat{d}} \mathbf{1}_{\{(d-1) \cdot T^d + t' \leq t-v\}} a_{st'} \leq \sum_{t'=1}^t y_{st} \leq \sum_{t'=1}^{t+v} \sum_{d=1}^{\hat{d}} \mathbf{1}_{\{(d-1) \cdot T^d + t' \leq t+v\}} a_{st'}$$

On the one hand, the number of starting shifts of pattern s in period t has to be at least as large as the number of shifts of pattern s scheduled until period $t - v$. Hence, no shift is postponed more than v periods. On the other hand, no shift is allowed to be pushed up by more than v periods. Thus, the number of shifts of pattern s started until period t cannot exceed the number of shifts of the pattern scheduled until period $t + v$. The concrete assignment of shifts to drivers can be done in a

Fig. 2 Example for cumulated starting times



post-processing step by assigning the next shift starting to the next unassigned shift of a driver with the corresponding shift pattern. Thanks to Constraints (15) and (16) the number of starting shifts of pattern s is not fixed any more but inside a corridor. This is also the reason why y_{st} has to be defined as an integer variable now (compare Constraints (17)).

Figure 2 gives an example for the cumulative starting times with v equals 30 min. The figure shows that the real starting times do not have to be equal to the scheduled ones. For example at 6:30am all drivers scheduled at 7am, who have to start latest at 7:30am, are already in action.

4.3 Additional shifts

A month’s demand structure can change significantly in comparison of two months. Reasons are public holidays or a general higher transportation demand in leisure times in summer in comparison to winter. Our so far constructed model cannot cope with this issues because the total amount of shifts is constant in each month. Thus, this section introduces the usage of working-time accounts to allow further shifts for drivers within a month. Therefore, we need to adapt Constraints (6) to adapt the number of active shifts x_{st} of shift pattern s in period t . We replace Constraints (6) by

$$y_{st} = \sum_{d=1}^{\hat{d}} a_{s,t-(d-1) \cdot T^d} + f_{s,t+T^d}^- + f_{s,t-(\hat{d}+1) \cdot T^d}^+ \quad \forall t \in T, s \in S. \quad (18)$$

While the first part ($\sum_{d=1}^{\hat{d}} a_{s,t-(d-1) \cdot T^d}$) counts the number of scheduled shifts of pattern s starting in period t (compare Constraints (5)), the second part ($f_{s,t+T^d}^-$) adds

additional shifts on the day before the first scheduled shift in the week and the third on the day after the last scheduled shift in the week ($f_{s,t-(\hat{d}+1) \cdot T^d}^+$). The sum of all of them equals the number of starting shifts y_{st} of pattern s in period t .

Constraints (18) do not restrict the extent of added shifts. Thus, they allow a driver who works on five days a week to work the entire week with one additional shift on the day before and one on the day after its regular working days. Therefore, the following constraints restrict the number of changes in the tour of a driver to h^s

$$\sum_{t': t=(t'+T^d) \bmod T^w} f_{st'}^- + \sum_{t': t=(t'-T^d \cdot \hat{d}) \bmod T^w} f_{st'}^+ \leq h^s \cdot a_{st} \quad \forall t = 1, \dots, T^w, s \in S \tag{19}$$

Thereby, *mod* is the modulo function such that all periods are pictured on their reference period in the first week of the time horizon. The left side counts all added shifts of pattern s starting the tour in period t . The right side bounds these shift changes to the number of started tours of the pattern in the period (a_{st}) times the number of allowed changes per driver (h^s).

However, if $h^s \geq 2$ a driver who works on five days a week can still have 12 working days in a row. The following constraints ensure that there is a solution with at least one day-off a week for each driver by restricting the number of added shifts by the number of scheduled shifts

$$f_{st}^- + f_{st}^+ \leq a_{st} \quad \forall t \in T, s \in S. \tag{20}$$

Thus, we can easily distribute the additional shifts to drivers in a post-processing step to ensure that each driver has not more than one additional shift per week by assigning at most one of the additional shifts f_{st}^- and f_{st}^+ to a driver.

In a practical application, there must be a balance in the working-time account over a year. As mentioned at the beginning of the section, ridepooling providers observe an increased transportation demand in leisure times in summer in comparison to winter and a varying customer behaviour on public holidays. Knowing these aspects makes it easy for the provider to identify days/months of higher or lower demand such that the usage of additional shifts can be steered by fixing f_{st}^- as well as f_{st}^+ variables to 0 if a lower demand is expected. Moreover, h^s can be set accordingly to steer the usage of additional shifts. We do not include the option to omit shifts to balance the working-time account in months with lower demand, as a shift of a full-time employed driver on a given day can be omitted for a driver starting its tour on this day or for a driver starting its tour on one of the last four days. Thus, a lot of symmetries accrue, which make the problem harder to solve. Instead, we suggest to define shifts with additional days-off in months with lower demand, which are assigned to drivers with a positive working-time account, or to omit shifts in a post-processing step in these months.

If we combine flexible starting times with the additional shifts, we add (17), (19)–(20), and the following two sets of constraints, which replace (6), (15), (16), and (18) to the basic model

$$\sum_{t'=1}^t y_{st'} \geq \sum_{t'=1}^{t-v} \left(\sum_{d=1}^{\hat{\Delta}} \mathbf{1}_{\{(d-1) \cdot T^d + t' \leq t-v\}} a_{st'} \right) + \mathbf{1}_{\{t' \leq t-v+T^d\}} f_{st'}^- + \mathbf{1}_{\{t' \leq t-v-(\hat{\Delta}+1) \cdot T^d\}} f_{st'}^+ \quad (21)$$

$\forall t \in T, s \in S$

$$\sum_{t'=1}^t y_{st'} \leq \sum_{t'=1}^{t+v} \left(\sum_{d=1}^{\hat{\Delta}} \mathbf{1}_{\{(d-1) \cdot T^d + t' \leq t+v\}} a_{st'} \right) + \mathbf{1}_{\{t' \leq t+v+T^d\}} f_{st'}^- + \mathbf{1}_{\{t' \leq t+v-(\hat{\Delta}+1) \cdot T^d\}} f_{st'}^+ \quad (22)$$

$\forall t \in T, s \in S.$

4.4 Model improvement

In this section, we present techniques to improve the formulation of the basic model by reducing the number of integer variables.

Having a closer look on Constraints (5) and (6), which determine x_{st} variables in dependency of a_{st} variables, and Constraints (4), which fix x_{st} variables and therefore the supply to 0 in closing hours, indicates that also some a_{st} variables can be fixed to 0. If period t' is in a closing hour and $I_{tt'}^s = 1$, (4)–(6) lead to $a_{st} = 0$. Thus, we can fix $a_{st} = 0$ if the shift would be active in a closing hour.

Furthermore, Constraints (9) and (10) ensure a correct assignment of tours and therefore drivers to depots. In Constraints (9), the drivers of shift s starting in period t are split across the depots. Constraints (10) ensure that the depots' capacity is hold. Note that the model only generally restricts the number of drivers per depot, not their distribution over shift patterns. However, if we sum up all Constraints (9), we get

$$\sum_{b \in B} \sum_{s \in S} \sum_{t=1}^{T^w} g_{st}^b = \sum_{s \in S} \sum_{t=1}^{T^w} a_{st}. \quad (23)$$

If we do the same with Constraints (10), we get

$$\sum_{b \in B} \sum_{s \in S} \sum_{t=1}^{T^w} g_{st}^b \leq \sum_{b \in B} cap_b. \quad (24)$$

Combining both leads to

$$\sum_{s \in S} \sum_{t=1}^{T^w} a_{st} \leq \sum_{b \in B} cap_b \quad (25)$$

which has to be fulfilled, as otherwise (23) and (24) cannot have a feasible solution. But if (25) is fulfilled, we can easily find a solution fulfilling (13), (23), and (24) by applying the procedure in Algorithm 1.

Algorithm 1 Assignment of tours to depots

```

1: Set  $g_{st}^b = 0$  for all  $s \in S$ ,  $t = 1, \dots, T^w$ , and  $b \in B$ 
2: for all  $t = 1, \dots, T^w$  do
3:   for all  $b \in B$  do
4:     Set  $g_{st}^b = \frac{cap_b}{\sum_{b' \in B} cap_{b'}} \cdot a_{st}$ 
5:   end for
6: end for

```

Note that Algorithm 1 assigns the tours evenly to the depots both in total and for each time period. Although this is not directly required, access roads to depots are typically restricted in practice such that it is beneficial to have smoothed departure times. This is also in line with Constraints (11). Note, however, that it might also be beneficial to use another distribution in Line 4 of Algorithm 1 for example if there is a higher demand to be expected in the vicinity of a depot at a certain time. Then, we could adapt the factor $\frac{cap_b}{\sum_{b' \in B} cap_{b'}}$ to change the assignment in the corresponding periods.

If we insert $g_{st}^b = \frac{cap_b}{\sum_{b' \in B} cap_{b'}} \cdot a_{st}$ on the left side of (23), we get

$$\sum_{b \in B} \sum_{s \in S} \sum_{t=1}^{T^w} \frac{cap_b}{\sum_{b' \in B} cap_{b'}} \cdot a_{st} = \sum_{s \in S} \sum_{t=1}^{T^w} \frac{\sum_{b \in B} cap_b}{\sum_{b' \in B} cap_{b'}} \cdot a_{st} = \sum_{s \in S} \sum_{t=1}^{T^w} a_{st}. \quad (26)$$

Thus, (23) is fulfilled and (24) is fulfilled if (25) is. However, the left side of (25) comprises all drivers of all shifts and is fulfilled (compare (6)) if

$$\sum_{s \in S} q_s \leq \sum_{b \in B} cap_b, \quad (27)$$

is fulfilled, which is a restriction to data. Indeed, it would not be useful to have in total over all shifts more active drivers (excluding those on holidays or being certified unfit for work) than the whole capacity of depots because not all of them could be deployed. Hence, we can assume that our data fulfils (27). Thus, we can assume that (25) is fulfilled such that we apply Algorithm 1 in the post-processing to find an assignment of drivers to depots (round accordingly if g_{st}^b is not integer in Line 4). Therefore, we can exclude Constraints (9), (10), and (13) from the model.

4.5 Post-processing steps

Beside the already mentioned points there are further questions which can be solved in post-processing steps. The model does not assign shifts to concrete drivers within the planning horizon. Variables a_{st} lead in combination with Constraints (8) to driver tours for the whole planning horizon. Hence, we only have to assign one driver to each tour at the beginning of the planning horizon. If flexible starting times are possible, concrete starting times can be determined as described in Sect. 4.2 (shift with the next scheduled starting time (a_{st} variables) is assigned to the next unassigned shift of a driver with the same pattern s). This assignment is fair, as it minimizes the total starting time variation over the planning horizon for all drivers.

Additional shifts with impact on the working-time account can also be distributed fairly within the planning horizon by assigning the next additional shift to the driver of the corresponding shift pattern and starting time with the smallest number of added shifts so far. As an alternative the next additional shift can be assigned to the driver with the most missing hours or the least overtime (historical), respectively.

Furthermore, our model does not include a transition between a scheduled time horizon and the next (e.g. months). Given the scheduled shifts of the first time horizon and those of the second, a minimum weight matching problem can be solved for each shift pattern to minimize the total variation of starting times over all drivers between two consecutive planning horizons by using the absolute difference between the scheduled starting times of the corresponding shifts of both planning horizons as weights.

5 Computational study

In this section, we first describe our instance generation by a cubic spline interpolation between peak and low points (Sect. 5.1). Then, we use the generator to evaluate our model on realistic instances in Sect. 5.2.

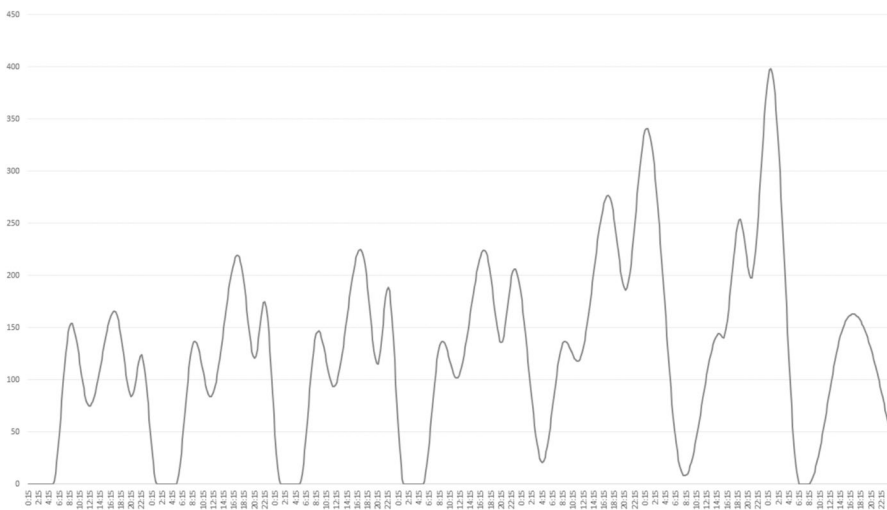
5.1 Generation of problem instances

The purpose of the computational study is to analyse the effectiveness of additional scheduling flexibility on the background of typical demand data for ridepooling services in Germany. Given that ridepooling is comparatively young and might have just left the "early adoption" stage (for a discussion see Kistorz et al. (2021)), demand structures can be bounded to change, so that such an analysis can only ever be temporarily. In what follows, we will first seek to capture a characteristic demand profile and then subject it to stochastic perturbation to generate a set of test instances. In deriving the demand profile, we will rely on observations published in a survey on Europe's leading ridepooling provider MOIA (see Kistorz et al. (2021)) and enrich it with some assumptions about relative demand between weekdays.

From a theoretical point of view, the operational utilization of capacity in service systems can be modelled as an inhomogeneous poisson process where an intensity

Table 2 Expected peak and low times

Time	Mo	Tu	We	Thu	Fri	Time	Sa	Time	Sun
04:15am	0	0	0	0	25	06:15am	45	02:00pm	120
08:00am	135	135	135	135	135	03:00pm	160	12:00pm	50
11:30am	90	90	90	90	110	03:30pm	150		
05:45pm	165	190	205	220	270	06:45pm	280		
08:30pm	100	115	115	130	160	09:00pm	205		
10:15pm	130	170	180	200	220	00:30am	400		

**Fig. 3** Example for weekly demand profile

function describes the expected value of vehicle requests over time and thus of utilized vehicles per period. Given that operational shift planning is typically carried out on demand forecasts, the process of finding appropriate demand estimates can be understood as capturing the intensity function of vehicle requests which will depend on aggregate client preferences and behaviour as well as external influences such as weather conditions, holidays, events, etc. In modern theoretical contributions and practical applications, demand estimates are often derived using machine learning techniques on the basis of past data on client behaviour. It is beyond the scope of this work to replicate such a demand forecast for a given period of time, nonetheless such an effort might also not be necessary as long as a characteristic profile can be identified that will tend to be predicted by any machine learning technique.

For instance, Kistorz et al. (2021) report the results of a survey with over 6000 respondents and derive several general insights about the client behaviour which drives hourly demand. In particular, the demand function on the weekdays (Mo-Fri) will typically be defined by three distinct peaks, one in the morning around 8am,

Table 3 Capacity scenarios

Scenario	(4, 5) and (5, 4)	(7, 1)
300	150	150
350	175	175
400	200	200

another one in the evening around 6pm, and one in the night time around 10pm (compare with Kistorz et al. (2021)). The extent of the peaks will depend on various factors such as the market penetration, the month, weather conditions, etc., but service demand tends to increase towards the end of the week, in particular for evening and night time use. Demand on Saturday is also characterized by three peaks, which occur much later on the day, starting at around 3pm and then at around 7pm and 0:30am where demand typically reaches its maximum over the week. Service demand will generally be lower on Sundays and less pronounced, while demand typically peaks in the afternoon around 2pm.

In order to determine a characteristic demand profile, we first seek to approximately capture the (local) peak and low points of the demand function and will then use a cubic spline interpolation to derive demand values for all other periods. For this purpose the month (four weeks) is divided into 15 min intervals. We model the demand function for a service provider who has a peak demand of approximately 400 vehicles which is reached on in the night on Saturday. For the typical local minima and maxima of the weekly demand the maximum is proportionally decreased as given in Table 2.

In order to derive varying problem instances the peak demand values are subjected to random perturbation with a random normal variable with mean 0 and a coefficient of variation (CV) which we set to 0.1, 0.2, and 0.3, respectively. We then use cubic spline interpolation to determine demand estimates for all values in the service time frame. We derive in total 30 planning instances in this way. A characteristic example profile is provided in Fig. 3.

For the supply side we assume three types of workers: Full time workers who work a shift of eight hours a day for five consecutive days a week. Part time workers who can work either four hours daily for five consecutive days a week, or five hours daily for four consecutive days a week and floaters who work a single seven hours on a given day. We distinguish three supply scenarios with 150, 175, and 200 part time drivers and floaters, respectively, to investigate the effect of different staffing mixes (compare Table 3). As our instances have a strongly varying demand due to the different CV, we do not restrict the number of full time drivers. Thus, Constraint (7) is omitted for $s = (8, 5)$.

In order to capture floaters and part time workers, the model has to be slightly adapted. Constraint (7) is only enforced for floaters. For part time workers it holds:

$$\sum_{s=(4,5) \text{ or } (5,4)} \sum_{t=1}^{T^w} a_{st} \leq p \tag{28}$$

where p is the number of part time workers. The remaining model parameters were set to: $c_t^o = 1$, $c_t^u = 2$, $h^s = 2$, $r^{max} = 50$, $T^d = 96$, $T^w = 672$, $v = 4$, and $|T| = 3072$ whereat the first 384 periods ensure a rolling horizon and are not counted in the objective function. The remaining 2688 periods picture a four week time horizon. All computational experiments were performed by GAMS-CPLEX (version 42) executed on a single AMD EPYC 7542 32 core with 2.90GHz. The maximum allowed runtime was 7200 CPU seconds per instance.

In order to study the influence on flexibility, we test three model variants. The *basic* model represents a planning approach without flexibility in the start times of shifts or working time accounts. It is comprised of the following constraint set: (1)–(6), (7) (only for floaters), (8), (11)–(12), (14), (28). The *flex* model allows for a variation of daily start times of shifts within certain limits and is comprised of the following constraints: (1)–(5), (7) (only for floaters), (8), (11)–(12), (14)–(17), (28). The working time model (*wTime*) introduces the possibility to work on additional shifts if the demand requires it and makes use of the following constraints: (1)–(5), (7) (only for floaters), (8), (11)–(12), (14), (18)–(20), (28).

Finally, the combined model *wTimeFlex* allows both flexible start times of shifts between days and an additional shift for working-time accounts: (1)–(5), (7) (only for floaters), (8), (11)–(12), (14), (17), (19)–(22), (28).

5.2 Discussion of results

In a first experiment, we seek to study the impact of start time flexibility on the solution performance. Table 4 shows the computational results for each of the capacity scenarios and compares the *basic* model without flexibility against an increasing possibility to vary daily start times of shifts by up to $v = 4$ (an hour), $v = 8$ (two hours), and $v = 12$ periods (three hours) in each direction. For *basic* all instances were solved to optimality in less than 120 s. For *flex4*, *flex8*, and *flex12* some instances could not be solved to optimality within 7200 s. However, optimality gaps are extremely low for the most part (less than 0.01% on average). It can be clearly seen that adding start time flexibility allows for a considerably better matching of supply and demand with relative oversupply cut from 15.5% to 12.2% for *flex12* on average and even more importantly undersupply being cut from 20.5% for *basics* model to 12.6% for *flex4* to 6.6% for *flex8* and to only 5.0% for *flex12*, even though there are clearly diminishing returns since *flex8* manages to come quite close to the results of *flex12* with only two third of the allotted flexibility.

If a more flexible mix of shift profiles is supplied as in the scenarios with 350 and 400 part time drivers, all approaches benefit in their solution quality even though computational effort continuously increases.

In order to investigate the potential of additional flexibility even further, Table 5 shows the performance of the model with working time accounts (*wTime*) and the combined models which include both, working time accounts and starting time flexibility (*wTimeFlex4*, *wTimeFlex8*, and *wTimeFlex12*). While near optimal solutions can be found for all instances of the working time model *wTime* within 7200 s, the combined models struggle more and more with increasing flexibility.

Table 4 Experimental results of basic and flexible models

part time	CV	<i>basic</i>				<i>flex4</i>				<i>flex8</i>				<i>flex12</i>			
		gap	cpu	under / over		gap	cpu	under / over		gap	cpu	under / over		gap	cpu	under / over	
300	avg	0.00%	31	21.4% / 15.9%	< 0.01%	2266	13.5% / 13.9%	< 0.01%	3140	7.2% / 14.5%	0.01%	5869	5.4% / 13.3%				
	0.1	0.00%	36	17.1% / 13.2%	< 0.01%	3533	10.2% / 10.7%	< 0.01%	4816	4.4% / 11.7%	0.03%	7200	3.0% / 10.6%				
	0.2	0.00%	33	19.9% / 16.8%	< 0.01%	1778	12.7% / 14.4%	0.00%	2187	7.1% / 14.5%	0.01%	5420	5.4% / 13.5%				
	0.3	0.00%	23	27.0% / 17.8%	< 0.01%	1487	17.6% / 16.7%	< 0.01%	2416	10.2% / 17.4%	< 0.01%	4986	7.9% / 15.8%				
350	avg	0.00%	33	20.4% / 15.5%	< 0.01%	3362	12.6% / 13.3%	< 0.01%	4298	6.6% / 13.5%	0.04%	6157	5.0% / 12.2%				
	0.1	0.00%	50	16.3% / 12.6%	< 0.01%	5959	9.2% / 10.1%	0.01%	6524	3.8% / 10.4%	0.09%	7200	2.6% / 9.3%				
	0.2	0.00%	29	19.1% / 16.3%	0.00%	2181	11.9% / 13.9%	< 0.01%	3024	6.4% / 13.5%	0.02%	6088	4.9% / 12.4%				
	0.3	0.00%	19	26.0% / 17.5%	< 0.01%	1945	16.8% / 16.0%	< 0.01%	3347	9.5% / 16.5%	0.02%	5184	7.4% / 14.9%				
400	avg	0.00%	38	19.6% / 15.1%	< 0.01%	4519	11.8% / 12.8%	1.61%	4991	6.2% / 12.6%	0.08%	6859	4.6% / 11.2%				
	0.1	0.00%	43	15.5% / 12.0%	0.01%	6944	8.5% / 9.5%	4.83%	6796	4.0% / 9.3%	0.20%	7200	2.3% / 8.2%				
	0.2	0.00%	41	18.3% / 15.9%	< 0.01%	3858	11.1% / 13.3%	< 0.01%	4275	5.6% / 12.8%	0.02%	7200	4.4% / 11.3%				
	0.3	0.00%	31	24.9% / 17.3%	< 0.01%	2756	15.9% / 15.5%	< 0.01%	3901	8.9% / 15.8%	0.02%	6177	6.9% / 14.1%				
avg	0.00%	34	20.5% / 15.5%	< 0.01%	3382	12.6% / 13.3%	0.54%	4143	6.6% / 13.6%	0.04%	6295	5.0% / 12.2%					

Table 5 Experimental results for working time and combined models

part time	CV	<i>wTime</i>			<i>wTimeFlex4</i>			<i>wTimeFlex8</i>			<i>wTimeFlex12</i>		
		gap	cpu	under / over	gap	cpu	under / over	gap	cpu	under / over	gap	cpu	under / over
300	avg	< 0.01%	1786	17.1% / 15.5%	2.59%	6586	10.5% / 12.1%	0.14%	6948	5.5% / 10.9%	14.66%	7200	6.9% / 10.2%
	0.1	0.00%	1576	13.2% / 12.7%	4.95%	7200	7.8% / 9.2%	0.12%	7200	3.2% / 7.9%	26.55%	7200	6.4% / 8.2%
	0.2	0.00%	1559	16.0% / 16.4%	0.15%	6725	9.6% / 12.3%	0.09%	6963	5.4% / 11.3%	5.85%	7200	5.5% / 9.9%
	0.3	< 0.01%	2224	22.0% / 17.5%	2.68%	5834	14.1% / 14.7%	0.21%	6683	8.0% / 13.5%	11.58%	7200	8.8% / 12.4%
	avg	< 0.01%	2441	16.0% / 14.9%	1.18%	6916	9.3% / 11.2%	5.43%	7051	5.7% / 9.9%	19.01%	7200	7.6% / 9.6%
350	0.1	0.00%	2477	12.4% / 11.8%	0.26%	7200	6.0% / 8.2%	6.51%	7200	3.2% / 7.1%	23.05%	7200	5.4% / 7.4%
	0.2	< 0.01%	1778	15.0% / 15.8%	0.08%	7012	8.6% / 11.5%	5.46%	7200	5.6% / 10.2%	6.94%	7200	4.9% / 9.0%
	0.3	< 0.01%	3068	20.6% / 17.1%	3.21%	6534	13.1% / 13.9%	4.32%	6754	8.2% / 12.4%	27.04%	7200	12.4% / 12.4%
	avg	< 0.01%	2676	15.0% / 14.4%	3.10%	7025	8.7% / 10.5%	14.64%	7099	6.4% / 9.5%	40.64%	7200	9.8% / 10.1%
	0.1	< 0.01%	3229	11.5% / 11.2%	6.12%	7200	6.2% / 7.8%	29.43%	7200	5.9% / 7.7%	77.53%	7200	12.6% / 10.0%
400	0.2	< 0.01%	1685	14.1% / 15.3%	0.12%	7200	7.9% / 10.6%	7.54%	7200	5.3% / 9.7%	19.42%	7200	6.7% / 9.1%
	0.3	< 0.01%	3113	19.4% / 16.8%	3.07%	6675	12.0% / 13.3%	6.95%	6897	8.0% / 11.2%	24.96%	7200	10.1% / 11.2%
	avg	< 0.01%	2301	16.0% / 15.0%	2.29%	6842	9.5% / 11.3%	6.74%	7033	5.9% / 10.1%	24.77%	7200	8.1% / 10.0%

The consideration of extra shifts in working time accounts decreases undersupply considerably by 4.5 percentage points with respect to the base model, but when compared with starting time flexibility, the solution quality of *wTime* is considerably worse. Among the combined models, the optimality gaps of *wTimeFlex4* and *wTimeFlex8* are still acceptable, whereas *wTimeFlex12* leads to higher gaps. Because of that the average performance of *wTimeFlex12* deteriorates and *wTimeFlex8* provides the best average performance. In comparison to the flex models in Table 4, additional working time helps to improve the results for $\nu = 4$ and $\nu = 8$. For $\nu = 12$ the results are worse due to the significant gaps for *wTimeFlex12*.

Given that the combined models could not reliably be solved to optimality, we conducted a final set of experiments where we used a step-wise method to attain shift schedules. In the first step, we used one of the core models (*basic*, *flex12*, and *wTime*), to find suitable columns for the shift plan and then in the next step these columns (a_{st} variables) were fixed and variable starting times ($\nu = 12$) and extra shifts were planned for these fixed columns. The results of these experiments are shown in Table 6. All instances could be solved to near optimality within 7200 s of CPU time (only second step). The use of *flex12* columns leads to the highest computation times in the second stage, but this additional computation time pays off, since these columns by far lead to the best solution performance. Keep in mind that undersupply is penalized twice as strong as oversupply and *flex12* columns achieve an average undersupply of only 3.4% compared to 9.4% and 9.5% for the *basic* and the *wTime* columns, respectively. As before a more flexible shift profile mix improves the performance of all approaches consistently, in particular for *basic* columns and *wTime* columns.

Over all three experiments we could see that an increased number of part time drivers improves the solution quality with the exception of *wTimeFlex8* and *wTimeFlex12* in Table 5 for which we could not close the gap sufficiently within the time limit of 7200 s. In the same way, solution performance decreases for all our models if the coefficient of variation increases (with the same exceptions). To give a visual impression of the solution quality, Figs. 4 and 5 display the coverage of the demand curve for an instance with a low CV of 0.1 (Fig. 4) and with a high CV of 0.3 (Fig. 5) and with 400 part time drivers. In both figures, the results of the *basic* model are compared with the best result (*flex12* columns). As can be seen in Fig. 4 the inflexible *basic* model cannot cover the demand on the highest peaks and night times on weekends while there is noteworthy oversupply on Mondays to Fridays. In contrast, the *flex12* columns model can cover the demand on weekends almost perfectly. However, the flexibility of the driver mix is still not sufficient, as there is some oversupply on the remaining days.

For the higher CV of 0.3 Fig. 5 presents the results. Demand varies a lot more over the days and reaches a higher maximum and lower minimum (over 700 vs. under 450 in Fig. 4). Here the *basic* model struggles even more with the highest peaks and, moreover, leads to undersupply on other days as well as significant oversupply at some times on the other days. Although the coverage can strongly be improved by using the *flex12* columns model, the highest peaks can still not be covered and there is still significant oversupply on the remaining days. Thus, the

Table 6 Experimental results for stepwise models

part time	CV	basic columns			flex/2 columns			wTime columns		
		gap	cpu	under / over	gap	cpu	under / over	gap	cpu	under / over
300	avg	< 0.01%	566	10.6% / 11.0%	< 0.01%	2314	4.0% / 14.3%	< 0.01%	972	10.5% / 8.7%
	0.1	0.00%	41	7.6% / 9.0%	< 0.01%	4536	2.0% / 11.5%	0.00%	76	7.4% / 6.7%
	0.2	< 0.01%	804	9.5% / 11.6%	< 0.01%	917	4.1% / 14.5%	< 0.01%	786	9.4% / 9.5%
	0.3	< 0.01%	853	14.8% / 12.4%	< 0.01%	1489	5.8% / 16.9%	< 0.01%	2055	14.8% / 10.0%
350	avg	< 0.01%	1491	9.4% / 10.6%	< 0.01%	3746	3.4% / 13.3%	< 0.01%	1396	9.4% / 8.1%
	0.1	0.00%	92	6.5% / 8.4%	< 0.01%	6105	1.5% / 10.3%	0.00%	1064	6.6% / 5.9%
	0.2	< 0.01%	1849	8.4% / 11.1%	< 0.01%	2468	3.5% / 13.4%	< 0.01%	1513	8.4% / 8.8%
	0.3	< 0.01%	2534	13.3% / 12.1%	< 0.01%	2666	5.1% / 16.1%	< 0.01%	1611	13.3% / 9.5%
400	avg	< 0.01%	2976	8.3% / 10.1%	< 0.01%	4199	2.9% / 12.3%	< 0.01%	3021	8.4% / 7.5%
	0.1	< 0.01%	2928	5.6% / 7.9%	0.01%	5946	1.1% / 9.2%	< 0.01%	2152	5.7% / 5.3%
	0.2	< 0.01%	2260	7.3% / 10.7%	< 0.01%	4362	3.0% / 12.4%	< 0.01%	3872	7.5% / 8.2%
	0.3	< 0.01%	3741	11.9% / 11.9%	< 0.01%	2289	4.6% / 15.2%	< 0.01%	3039	12.0% / 9.0%
avg	< 0.01%	1678	9.4% / 10.6%	< 0.01%	3420	3.4% / 13.3%	< 0.01%	1796	9.5% / 8.1%	

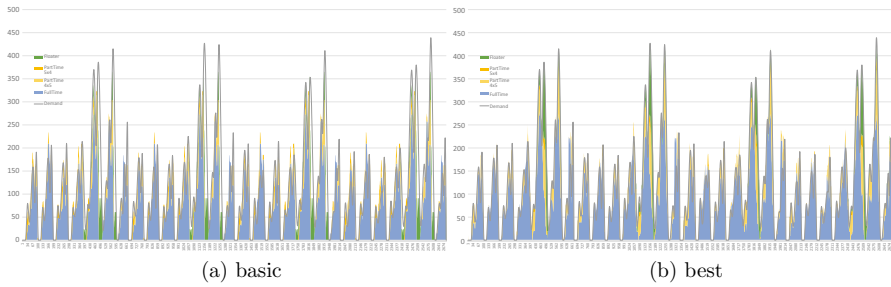


Fig. 4 Supply for an instance with low CV (0.1) and 400 part time drivers

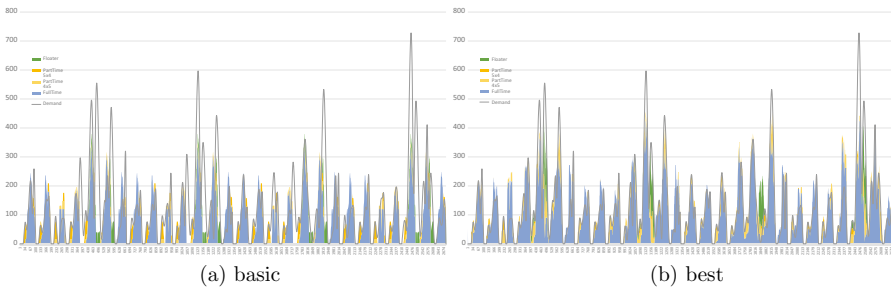


Fig. 5 Supply for an instance with high CV (0.3) and 400 part time drivers

flexibility (additional shifts, start time flexibility as well as shift mix) is not sufficient to face the strong demand variation.

6 Conclusion

This paper develops a shift scheduling model for ridepooling services which face strongly fluctuating demand over the week as well as between weeks. Therefore, small time periods (15 min) as well as flexible shifts are necessary to cover the demand. The proposed model uses flexibility in starting times and extra shifts, the extent of which both can be set by the operator weighing preferences between company and drivers. Moreover, the provider can evaluate different contract structures to determine the optimal shift mix for the company’s needs. Finally, the model can be used to schedule shift patterns that approximate the varying demand. The computational study demonstrates that the model is able to provide near optimal solutions for practice relevant sizes of large ridepooling services within two hours of computation time.

In future research, the introduced flexibility measures could be further extended for instance by considering shift extensions and omissions or break scheduling. A further interesting research direction is to incorporate the relationship between the availability of vehicles in the field and the demand for the service. In this work,

vehicle demand was considered an exogenous value, in line with the standard assumptions of ride pooling operators. In reality, additional available capacity can make the service more attractive, for instance due to lower response and waiting times, and thus lead to increased demand for service. The relationship between capacity availability and demand in service systems has for instance been investigated by incorporating queuing models with abandonment in shift scheduling models, see Defraeye and Van Nieuwenhuysse (2016). In the case of ride pooling services, this relationship is even more complex, since the ability to pool customer requests makes an estimation of required vehicles for increasing demand quite challenging (e.g. see the extensive study of Shulika et al. (2024)). Modeling demand endogeneity for ride pooling services might therefore constitute a formidable research project that can further improve decision support in shift scheduling.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest The authors declare that they have no Conflict of interest.

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

- Álvarez E, Ferrer J-C, Muñoz JC, Henao CA (2020) Efficient shift scheduling with multiple breaks for full-time employees: a retail industry case. *Comput Ind Eng* 150:106884
- Aykin T (2000) A comparative evaluation of modeling approaches to the labor shift scheduling problem. *Eur J Oper Res* 125(2):381–397
- Azmat CS, Widmer M (2004) A case study of single shift planning and scheduling under annualized hours: a simple three-step approach. *Eur J Oper Res* 153(1):148–175
- Brunner JO, Bard JF, Kolisch R (2009) Flexible shift scheduling of physicians. *Health Care Manage Sci* 12(3):285–305
- Burke EK, De Causmaecker P, Berghe GV, Van Landeghem H (2004) The state of the art of nurse rostering. *J Schedul* 7(6):441–499
- Carotenuto P, Giordani S, Salvatore A, Biasini A (2019) Resource planning for aircraft refueling in airport parking area. *Transp Res Proc* 37:250–257
- Cavada JP, Cortés CE, Rey PA (2020) A workforce planning and allocation model for the outbound baggage loading area at santiago international airport. *INFOR Inf Syst Oper Res* 58(3):537–559
- Corominas A, Lusa A, Olivella J (2012) A detailed workforce planning model including non-linear dependence of capacity on the size of the staff and cash management. *Eur J Oper Res* 216(2):445–458
- Corominas A, Olivella J, Pastor R (2010) Capacity planning with working time accounts in services. *J Oper Res Soc* 61(2):321–331
- Defraeye M, Van Nieuwenhuysse I (2016) Staffing and scheduling under nonstationary demand for service: a literature review. *Omega* 58:4–25

- El-Rifai O, Garaix T, Augusto V, Xie X (2015) A stochastic optimization model for shift scheduling in emergency departments. *Health Care Manage Sci* 18(3):289–302
- Erhard M, Schoenfelder J, Fügener A, Brunner JO (2018) State of the art in physician scheduling. *Eur J Oper Res* 265(1):1–18
- Ernst AT, Jiang H, Krishnamoorthy M, Sier D (2004) Staff scheduling and rostering: a review of applications, methods and models. *Eur J Oper Res* 153(1):3–27
- Henao CA, Muñoz JC, Ferrer JC (2010) The impact of multi-skilling on personnel scheduling in the service sector: a retail industry case. *J Oper Res Soc* 66:1949–1959
- Hertz A, Lahrichi N, Widmer M (2010) A flexible milp model for multiple-shift workforce planning under annualized hours. *Eur J Oper Res* 200(3):860–873
- Ho SC, Szeto WY, Kuo Y-H, Leung JM, Petering M, Tou TW (2018) A survey of dial-a-ride problems: Literature review and recent developments. *Transp Res Part B Methodol* 111:395–421
- Horn M, Frohner N, Raidl GR (2021) Driver shift planning for an online store with short delivery times. *Proc Comput Sci* 180:517–524
- Kabak Ö, Ülengin F, Aktaş E, Önsel Ş, Topcu YI (2008) Efficient shift scheduling in the retail sector through two-stage optimization. *Eur J Oper Res* 184(1):76–90
- Kostorz N, Fraedrich E, Kagerbauer M (2021) Usage and user characteristics—insights from Moia, Europe’s largest ridepooling service. *Sustainability* 13(2):958
- Kuehnel N, Zwick F, Hörl S (2021) Shifts in perspective: modeling operational challenges in non-autonomous and electric ride-pooling systems
- Ladier A-L, Alpan G, Penz B (2014) Joint employee weekly timetabling and daily rostering: a decision-support tool for a logistics platform. *Eur J Oper Res* 234(1):278–291
- Pandey P, Gajjar H, Shah BJ (2021) Determining optimal workforce size and schedule at the retail store considering overstaffing and understaffing costs. *Comput Ind Eng* 161:107656
- Pastor R, Olivella J (2008) Selecting and adapting weekly work schedules with working time accounts: a case of a retail clothing chain. *Eur J Oper Res* 184(1):1–12
- Pfeiffer C, Schulz A (2022) An alns algorithm for the static dial-a-ride problem with ride and waiting time minimization. *OR Spect* 44:87–119
- Pfeiffer C, Schulz A (2022) A new lower bound for the static dial-a-ride problem with ride and waiting time minimization. In: Freitag M, Kinra A, Kotzab H, Megow N (eds) *Dynamics in logistics*. Springer International Publishing, Cham, pp 231–243
- Restrepo MI, Semet F, Pocreau T (2019) Integrated shift scheduling and load assignment optimization for attended home delivery. *Transp Sci* 53(4):1150–1174
- Schulz A, Pfeiffer C (2024) Using fixed paths to improve branch-and-cut algorithms for precedence-constrained routing problems. *Eur J Oper Res* 312(2):456–472
- Shulika O, Bujak M, Ghasemi F, Kucharski R (2024) Spatiotemporal variability of ride-pooling potential—half a year new york city experiment. *J Transp Geogr* 114:103767
- Sillekens T, Koberstein A, Suhl L (2011) Aggregate production planning in the automotive industry with special consideration of workforce flexibility. *Int J Prod Res* 49(17):5055–5078
- Topaloglu S, Ozkarahan I (2004) An implicit goal programming model for the tour scheduling problem considering the employee work preference. *Ann Oper Res* 128:135–158
- Van den Bergh J, Beliën J, De Bruecker P, Demeulemeester E, De Boeck L (2013) Personnel scheduling: a literature review. *Eur J Oper Res* 226(3):367–385

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