Minimizing Delays of Patient Transports with Incomplete Information

Dennis Adelhütte\textsuperscript{1}, Kristin Braun\textsuperscript{*1}, Frauke Liers\textsuperscript{1}, and Sebastian Tschuppik\textsuperscript{1}

\textsuperscript{1}Friedrich-Alexander-Universität Erlangen-Nürnberg, Department Data Science, Cauerstraße 11, 91058 Erlangen

1\textsuperscript{st} December, 2021

Abstract

We investigate a challenging task in ambulatory care, namely minimizing delays in patient transport. In practice, a limited number of vehicles is available for non–emergency transport. Furthermore, the dispatcher rarely has access to complete information when establishing a transport plan for dispatching the vehicles. If additional transport is requested on demand, then schedules have to be updated, which can lead to long waiting times. We model the scheduling of patient transports as a vehicle routing problem with general time windows and solve it as a mixed–integer linear problem that is modified whenever additional transport information becomes available. We propose a modeling approach that is designed to determine fair and stable plans. Furthermore, we show that the model can easily be modified when transports need to satisfy additional requirements e.g. during the Covid–19 pandemic. To show the applicability and efficiency of our modeling approach, we conduct a numerical study using historical data from the region of Middle Franconia. The results reveal and show that, in general, mathematical optimization methods for mixed–integer linear programs can significantly decrease delays and have considerable potential for optimized patient transportation.

Keywords: OR in Health Services, Vehicle Routing, Heuristics, Mixed–Integer Linear Optimization, Patient Transport

1 Introduction

In a healthcare system, patients need to be transported to guarantee a functioning system. Since such a system comprises many components, for example health promotion, primary health care, specialist services and hospitals, transporting patients without delays is not trivial at all, especially when there is not an emergency. This could be the case when patient needs to be transported from a hospital to their home or patients receiving certain treatments at a specific destinations. While in principle those transports could be carried out by the same vehicles that carry out rescue transports, a different fleet is typically used for non–urgent patient transport in Germany. In contrast to emergency rescue, patient transports allow delays, though this is not desirable. Scheduling them is a challenge due to different reasons. While the number of vehicles in the transport fleet is limited, the drivers’ work shifts have to be respected. Moreover, a vehicle can transport only one patient at once. Finally, not all transportation requests are known at the time when the plans are established: Many transports are requested during the day when a transport schedule is already in operation. The last point in particular can lead to long delays for patients waiting for their transport, as it is often impossible to handle all transports at once.

Whenever a transport is requested, the vehicle that can reach the patient most quickly is dispatched. This scheduling approach is also performed in Middle Franconia, according to the local dispatcher.\textsuperscript{1}

\textsuperscript{*}Corresponding author

\textsuperscript{1}Integrated Control Center Nuremberg (ILS)
For the resulting transport plans, optimization potential is usually disregarded and sometimes, transports have even to be rescheduled to the following day. As a motivation, we provide some statistics about the transport data in the following.

Statistics about the patient transports  We consider the number of transports from January 2020 to July 2021. In Figure 1a, we plot the number of patient transports. There are drops in the number of transports that occur each week corresponding to the weekends, especially to Sundays, while the overall number of transports has a similar magnitude. One can also see that over the turn of the year, due to Christmas and other holidays, less transports are requested. In 2020, there is another small drop starting in April. In previous years there is no such drop, so this is probably caused by the beginning of the Covid–19 pandemic. In Figure 1b, we plot the percentage of transports that involved an (at least suspected) patient with Covid–19. These are represented by the black portion of the bars. For better visibility, the percentage of Covid–19 transports is also given in Figure 1b. In the peak phase up to 60% of all transports were classified as infected cases. This is – as there are a lot of suspected cases – not the actual number of infections. However, the trend given in Figure 1b is quite similar to the number of actual Covid–19 cases in Middle Franconia, cf. [27]. This trend is given in the red part of the plot.

Furthermore, we consider the number of transports during the course of a day in Figure 2: Transports are distinguished depending on their target time and the number of requested transports per hour is plotted for the same time window as before. The majority of the transports are requested at 6 am or later. The peak is in the late morning, thereafter the number slowly decreases. After about 7 pm, the number of transports is again relatively low compared to the previous hours. The number of plannable transports is given in the gray bars, while the black bars represent the number of ad–hoc transports. In the early morning hours, i.e., between 6 and 8 am most transports are ad–hoc ones. After that, until noon, most transports are plannable. Then, the percentage of ad–hoc transports increases over the course of the day. After 3 pm, more than 80% of all transports are ad–hoc transports.

Our contribution  In this paper, we model the problem of finding fair schedules for patient transports: In our case, 'fair' means that the maximum delay in all patient transport is minimized as the first priority, before, secondly, the total delay over all transports is minimized. Besides the aim of distributing the minimal delay among the patients in a roughly equal fashion, we have the requirement that the drivers’ shift times are not violated whenever possible. We present two approaches to handle transports that are requested during the course of the day: If the
dispatcher has knowledge of the requested transport in advance, but does not yet know the time it is supposed to happen (for example a patient needs to be taken home after treatment), then a data-driven approach is used to estimate that time. Otherwise, we re-optimize the transportation that has not yet begun or introduce so-called dummy transports that model artificial future transports that can be expected due to, for example, requests that typically come in at specific times of the day or as follow-up transports. The model is based on the vehicle routing problem with general time windows (VRPGTW) that was introduced in [18]. We solve the optimization problems using state-of-the-art solvers for mixed-integer linear programs that we enhance with heuristics and branching in order to improve their performance. We demonstrate that one can modify our proposed model whenever necessary by introducing a general and adaptable way to add new inequalities, equations and penalty variables. As an application, we discuss the additional issues that have to be taken into account during the Covid-19 pandemic to minimize the risk of infection and how to model them. In our numerical study, we show that our optimized procedure for scheduling transports is effective and efficient in practice. Regarding infectious illnesses, we show that transporting patients that are (at least suspected to be) infected using separate vehicles is desirable.

**Structure of the paper** In Section 2, the patient transport problem is described on an abstract level. We first define different transports and discuss the amount of information available at the time of planning. We also introduce model techniques for modifying models on an abstract level. In Section 3, the vehicle routing problem with general time windows is introduced. We discuss how the patient transport problem can be modeled as a mixed-integer linear problem (MIP). Subsequently, the model is modified to incorporate the required updates when a previously unknown transport is requested. We then describe how to incorporate Covid-19-related requirements, including disinfection time and the goal of separating known Covid-19 transports from other requests as far as possible.

In Section 4, we elaborate on algorithmic methods to improve the running time for our models’ solving process and evaluate the methods by using the available historical data. In particular, we compare our methods to an implementation simulating the current scheduling practice at the ILS. Finally, in Section 5, we discuss our results and provide some ideas for future research.

**Literature review** For a general overview of vehicle routing problems that were presented for the first time in [11], we refer to [29]. There, an overview of different methods for modeling and solving variations of vehicle routing problems and its modifications are presented and evaluated, including
branch–and–bound–algorithms, branch–and–cut–algorithms, set–covering–based algorithms and heuristic methods, among others. Further, another very detailed overview over dynamic vehicle routing is given in a chapter of this book, namely [4]. Since our goal is to establish mixed-integer linear models based on the vehicle routing problem with general time windows, we use state–of–the–art software to solve the models instead of applying methodology for solving vehicle routing problems.

The work of [10] presents an overview of different problem classes. Two generalizations are the vehicle routing problems with time windows and the vehicle routing problem with pick-up and delivery. The latter one nearly always contains time windows, so one mostly omits the pick-up and delivery part. In our work, we use the VRPGTW which was introduced in [18]. Contrary to the vehicle routing problem with time windows, the bounds on target times can be soft, i.e., their violation is penalized, or hard, i.e., the corresponding constraints need to be satisfied. Another generalization explained therein is the dial–a–ride problem where passengers are transported, thus human factors like their satisfaction must also be observed. Vehicle routing problems are either static or dynamic and either deterministic or stochastic. In a static VRP, all transports are known beforehand, while in a dynamic one they can change over time. Further, new transports can be requested during the execution period. The distinction between deterministic and stochastic VRP depends on whether some inputs are random variables or not. In this work, we consider a dynamic and deterministic VRP.

In [5], an overview of dynamic vehicle routing problems is given. In the work of [24] the distinction between periodic and continuous solving methods for dynamic VRPs is presented and recent work is categorized. A periodic solving method re–optimizes the problem after a certain (fixed) time period or when new data is available while a continuous solving method is performed throughout the whole day. A more recent work ([26]) further introduces a new classification scheme that consists of eleven criteria, for example the type of the objective function (i.e. whether one minimizes costs, distances, travel times, etc.), the fleet size, the type of time constraints and the solution method.

Many applications consider order picking and delivery problems. In [34] and [33], the authors consider delivery for business–to–costumer and online–to–offline supermarkets, respectively. A more general work can be found in [8] where a VRP for minimizing the driven road distance is modeled and evaluated on benchmark sets. In [15], the main question is to decide whether ad–hoc transports should be accepted or rejected. During the whole time frame, the problem is optimized in a continuous way.

Further solution methods relevant to our work are given in [14]. There, the authors introduce dummies to precautionarily schedule ad–hoc transports to areas where new requests are likely to occur.

In [6], the authors apply the solution of snapshots to handle ad–hoc transports for online taxi routing. There is also a lot of work that apply vehicle routing approaches to solve passenger transporting problems that occur in healthcare. In [3] and [19], the patient transport problem within a hospital is considered. In the former one, they assume that the hospital is one building, while in the latter one, the buildings of the hospital are spread over the whole city – in this specific example, the city Tours in France. A static version of the VRP for the patient transportation problem is given in [22], an example for a work about emergency transports can be found in [16]. Important work in the field of patient transport in the European area is described in [28], where the focus is on the situation in the Netherlands. There, ambulances can also be used for patient transportation. This, however, reduces the number of vehicles available for rescue missions. Although in principle this is possible in Germany as well, this option is usually avoided by the dispatcher, and we do not consider this option in our work. In [12], the authors describe the situation in Austria where the settings differ from the one presented here, since they describe the stationing of rescue vehicles or the periodic delivery of blood reserves. Another very recent work, with data from the US, is given in [32]. They also consider the patient transport problem, but in a stochastic version. In their work, service duration and travel time can be stochastic. They also have random cancellations. They use a $K$–means clustering–based algorithm to solve the dynamic part of the problem.

Concerning additional requirements due to the Covid–19 pandemic, [23] describes a successful application of the vehicle routing problem in practice, which is solved heuristically. The authors used tabu search to heuristically plan the distribution of face masks in Spain. In [2], the authors
tested the feasibility of schedules for transporting dialysis patients under worst case assumptions for the spreading of the virus using Monte Carlo simulations. Contactless delivery of food to settlements during the pandemic was considered in [7] and solved by applying a genetic algorithm.

2 Description of the problem

A transport of a patient includes the following components: A person has to be driven from one place to another. A vehicle cannot transport more than one patient a time and each transport has an origin and a destination. It also has a certain duration (the travel time and other tasks that have to be carried out) and a target time. We formalize this:

Definition 2.1. A plannable patient transport $T$ is a tuple $(O_T, D_T, d_T, t_T)$ that consists of

i) its origin $O_T$, i.e., the place where a patient is picked up,

ii) its destination $D_T$, i.e., the place where a patient is dropped of,

iii) its duration $d_T$, i.e., the total duration of the transport from reaching $O_T$ until leaving $D_T$,

iv) its target time $t_T$, i.e., the time when $T$ is supposed to be started.

The set of plannable transports is denoted as $\mathcal{T}_{\text{plan}}$. The dispatcher has complete information about the set $\mathcal{T}_{\text{plan}}$ when scheduling the vehicles.

A vehicle always starts its tour at its depot and ends there after finishing its transports, independently of the actual schedule. Since in practice the vehicles’ drivers have work shifts, a tour should start and end during the shift. This requirement typically turns out to be a bottleneck as we will see in our numerical study. Thus, we define the following:

Definition 2.2. Let $\mathcal{T}_{\text{plan}}$ be a set of plannable patient transports and let $y_T \geq t_T$, $T \in \mathcal{T}_{\text{plan}}$, be the actual time when a transport $T$ has started. The set $\mathcal{P} := \{(T, y_T) : T \in \mathcal{T}_{\text{plan}}\}$ is called the schedule. The schedule is called feasible if

i) for two subsequent transports $i, j \in \mathcal{T}_{\text{plan}}$ carried out by the same vehicle, $y_j \geq y_i + d_i$ holds and

ii) each vehicle starts and ends its trip within its shift.

If only (i) holds, then $\mathcal{P}$ is called weakly feasible. The set of (weakly) feasible schedules is henceforth denoted as $\mathcal{X}$ ($\mathcal{X}_{\text{weak}}$).

In this definition, the delay of vehicles is not taken into account when it comes to feasibility, as the target time of a transport is not considered for feasibility. When delays are too high, in practice, they can also be scheduled to another day. However, the vehicles’ shift times must be adhered to for feasible schedules.

The two conflicting goals of keeping the shift times and minimizing delays in a fair way already raises the question of what constitutes a ‘good’ feasible schedule. To treat more than one objective, we introduce the following:

Definition 2.3. Let $\mathcal{X}_{\text{weak}}$ be a set of weakly feasible schedules and assume that we are given $l \in \mathbb{N} := \{1, 2, \ldots \}$ real functions $g_1(x), \ldots, g_l(x)$, all defined on $\mathcal{X}_{\text{weak}}$, to measure the quality of the schedules. Let $\gamma_1, \ldots, \gamma_l \geq 0$ be a set of weights. We call the function $f : \mathcal{X}_{\text{weak}} \to \mathbb{R}$ with $f(x) := \sum_{i=1}^{l} \gamma_i g_i(x)$ the measure of quality with respect to the weights $\gamma_1, \ldots, \gamma_l \geq 0$.

Remark 2.4. The method of weighing objectives to obtain exactly one objective is referred to as scalarization and is applied in multi–objective optimization. Since we use single–objective optimization methods, we simply refer to [13].
Possible criteria include the maximum delay of all transports and the sum of the delays of all transports or the violation of shifts. We now formulate the optimization problem on an abstract level: For a given set of transports and given quality criteria with weights, the task is to solve

$$\min_{x \in \mathcal{X}} f(x) \text{ or, if necessary, } \min_{x \in \mathcal{X}_{\text{weak}}} h(x).$$  \hspace{1cm} (1)

The exact model is given in Section 3. If we optimize over $\mathcal{X}_{\text{weak}}$, then we assume that the violation of shift times is one of the measurements of quality $g(x)$ of the schedule in the sense of Definition 2.3, resulting in $h(x) := f(x) + \gamma g(x)$ for some $\gamma \geq 0$.

Incomplete information can manifest itself in two ways: Firstly, a transport has been requested but its target time is not known. This can happen, for example, if a patient needs to be returned home after a treatment but the dispatcher does not know when it will be finished. Secondly, a transport is requested that was not known about when establishing a schedule for $\mathcal{T}_{\text{plan}}$. For example, this could be patients that are discharged from the hospital. In both cases, the transport requests have to be incorporated into the transport schedule. We formalize the different levels of information:

**Definition 2.5.** A **semiplannable patient transport** $T$ is a tuple $(O_T, D_T, d_T)$ that consists of

i) its origin $O_T$, i.e., the place where a patient is picked up,

ii) its destination $D_T$, i.e., the place where a patient is dropped off and

iii) its duration $d_T$, i.e., denotes the total time of the transport.

For a semiplannable transport, no target time is known. The set of semiplannable transports is denoted as $\mathcal{T}_{\text{semi}}$.

**Definition 2.6.** Let $\mathcal{P}$ be a (weakly) feasible schedule for $\mathcal{T}_{\text{plan}}$. An **unplannable** or **ad-hoc transport** $T$ is a transport that has to be planned after vehicles have already started carrying out $\mathcal{P}$. The set of unplannable transports is denoted as $\mathcal{T}_{\text{ad-hoc}}$.

The notion of plannable, semiplannable and unplannable transports and the different level of information leads to a natural split of the scheduling process into two parts, similar to what was suggested in [21] in the context of recoverable robustness. In the planning phase, the transports of $\mathcal{T}_{\text{plan}}$ are scheduled. After the planning phase, when the schedule is carried out, the operational phase begins. While the transports of $\mathcal{T}_{\text{plan}}$ are carried out, transports of $\mathcal{T}_{\text{ad-hoc}}$ have to be incorporated into the schedule and the schedule usually has to be updated. For transports $T \in \mathcal{T}_{\text{semi}}$, no target time is yet defined. When one treats them as ad–hoc transports, the dispatcher ignores them until their target time becomes known. Alternatively, its target time could be estimated, denoted as $t^\text{est}_T$ and it could be treated like a plannable transport with target time $t^\text{est}_T$, ultimately planning the transport together with $\mathcal{T}_{\text{plan}}$. However, the true target time may differ from its estimate. If the target time is earlier than estimated, the dispatcher will be informed and can still treat the request as an ad–hoc transport. If the target time is later than estimated, then the scheduled vehicle has to wait. Since the target time is only an estimated time, the actual target time could be much later, resulting in a vehicle having to wait while it could carry out other transports. Therefore, a waiting time is introduced. If the actual target time is at least its waiting time, the scheduled vehicle does not wait any longer. The corresponding transport is then treated as an ad–hoc transport. This is formalized as follows:

**Definition 2.7.** A transport $T$ is called a **dummy transport**, if

- $T \in \mathcal{T}_{\text{semi}}$, i.e., it is semiplannable,

- its target time is estimated with $t^\text{est}_T$ and

- it has a waiting time $w_T \geq 0$ after which $T$ will be treated as an ad–hoc transport.

The set of dummy transports is denoted as $\mathcal{T}_{\text{dummy}}$. 

6
To estimate the target time, historical data or medical expertise can be used. An example for a scenario where dummy transports can be used are return trips from dialysis. We come back to this in Subsection 3.3.

Now we introduce two additional concepts that can be used to model, if necessary, further requirements in the patient transport problem:

**Definition 2.8.** Let $\Lambda$ be a set of variables. A **penalty weight** is a parameter $\gamma \in \mathbb{R}$ that is used to penalize variables $\lambda \in \Lambda$. A **penalty set** $\Gamma$ contains all such tuples $(\gamma, \Lambda)$. Using this representation, a general objective function has the form

$$
\sum_{(\gamma, \Lambda) \in \Gamma} \sum_{\lambda \in \Lambda} \gamma \cdot \lambda.
$$

(2)

**Definition 2.9.** A *(transport) label* is a function $\phi: \mathcal{T} \rightarrow \{0, 1\}$ that indicates whether a transport $T \in \mathcal{T}$ fulfills some property. Multiple labels can be collected in a label set $\Phi$.

Label information may include whether a transport involves infectious diseases, has to fulfill certain priorities, or the question whether certain equipment is needed. Another example is the information about the type (plannable, semiplannable, ad–hoc, dummy) of a transport. A similar set can also be introduced for vehicles – and for example indicate whether certain equipment is available. In this case, only such vehicles can handle transports that need this equipment. Using these concepts, we can introduce additional constraints for the patient transport problem. These constraints can be hard or soft constraints, the latter using penalty weights and additional infringement variables. This concludes the discussion of the patient transport problem. In the next section, we model the specific optimization problems and describe how to incorporate semiplannable and unplannable transports into the model.

## 3 The model and its application to the patient transport problem

The task from the previous section is now modeled as a VRPGTW. We start with explaining the VRPGTW and how it can be modeled as a MIP. Then, we proceed with the modeling of the plannable phase of our patient transport problem in Subsection 3.2. The semiplannable and ad–hoc transports are discussed in the following Subsection 3.3 before we elaborate on the modeling of requirements during the Covid–19 pandemic in Subsection 3.4.

### 3.1 The vehicle routing problem with general time windows

The VRPGTW is defined on a directed graph $G := (V, A)$ with vertex set $V := \{0, \ldots, n\}$, $n \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$, arc set $A \subseteq V \times V$ and a set of homogeneous vehicles $K := \{1, \ldots, m\}$, $m \in \mathbb{N}$. For all $i \in V$, the set of outgoing arcs is $\delta^+(i)$ and the set of incoming arcs is $\delta^-(i)$. Each node $i \in \{1, \ldots, n\} =: N$ corresponds to one customer, while node 0 is the depot where all vehicles start and end their trip. The duration $\tau_{i,j} \geq 0$ denotes the time that is required to serve customer $i$ and to drive to customer $j$ afterwards.

For all $(i, j) \in A$ and $k \in K$, the binary variable $x_{i,j,k} \in \{0, 1\}$ indicates whether vehicle $k$ travels on the arc $(i, j)$. For all $i \in N$, $y_i \in \mathbb{R}_+: = \{x \in \mathbb{R} \mid x \geq 0\}$ is the time when customer $i$ is served. Finally, for each $i \in V$, $p_i: \mathbb{R}_+ \rightarrow \mathbb{R}_+$ is a piecewise linear penalty function which yields a penalty dependent on the delay necessary to serve customer $i$. The goal is serving each customer exactly once, with as little penalty as possible. The model can be described as a mixed–integer nonlinear optimization problem (MINLP) that is easy to linearize, see Model (3). Our objective is to minimize the penalties. Constraints (3b) – (3e) ensure that there is a tour where each vehicle starts and ends at the depot and each customer is served exactly once, see [29]. Together with Constraint (3f), valid tours are defined. It ensures that a vehicle which serves customer $j$ directly after customer $i$ finishes the service of customer $i$ and travels from customer $i$ to $j$ before the
vehicle starts serving customer \( j \):

\[
\min \sum_{i \in N} p_i(y_i), \tag{3a}
\]

subject to

\[
\sum_{k \in K} \sum_{j \in \delta^+(i)} x_{i,j,k} = 1 \quad \forall i \in N, \tag{3b}
\]

\[
\sum_{j \in \delta^-(i)} x_{0,j,k} = 1 \quad \forall k \in K, \tag{3c}
\]

\[
\sum_{j \in \delta^+(i)} x_{i,j,k} - \sum_{j \in \delta^-(i)} x_{i,j,k} = 0 \quad \forall k \in K, i \in N, \tag{3d}
\]

\[
\sum_{i \in \delta^-(0)} x_{i,0,k} = 1 \quad \forall k \in K, \tag{3e}
\]

\[
x_{i,j,k}(y_i + \tau_{i,j} - y_j) \leq 0 \quad \forall k \in K, (i, j) \in A, \tag{3f}
\]

\[
x_{i,j,k} \in \{0, 1\} \quad \forall k \in K, (i, j) \in A, \tag{3g}
\]

\[
y_i \geq 0 \quad \forall i \in N. \tag{3h}
\]

Linearizing piecewise linear functions and Constraint (3f) are standard techniques. In particular, Constraint (3f) is equivalent to

\[
y_i + \tau_{i,j} - y_j \leq M_{i,j}(1 - x_{i,j,k})
\]

for a sufficiently large real number \( M_{i,j} > 0 \).

Thus, the resulting model is a MIP that is NP-hard in theory. However, modeling the VRP as a MIP has two major advantages: Firstly, we can use off-the-shelf software for solving it to global optimality. Secondly, if required, the model can easily be modified without needing to apply different algorithms or oracles which we demonstrate in Subsection 3.2.

Remark 3.1. In [18], the vehicles also have a capacity for transporting goods, and other costs are involved as well. However, for our purposes, the given parameters are sufficient since our vehicles can only transport exactly one patient a time and we are only interested in minimizing delays of transports.

### 3.2 Modeling the patient transport problem as a VRPGTW

Let us assume that \( |\mathcal{T}_{\text{plan}}| = n \) plannable transports have to be scheduled. The instance of the VRPGTW is defined as follows:

- \( N := \{1, \ldots, n\} \) is the set of transports. The depot is denoted as node 0 and has a copy \( n + 1 \), the vehicles start at 0 and end their trip at \( n + 1 \) (this is done to address modeling issues and does not have other implications).
- \( K := \{1, \ldots, m\} \) is the set of available vehicles for transport.
- \( A_N := \{(i, j) \in N \times N : i \neq j\} \) is the set of arcs ‘between two transports’. An arc \((i, j) \in A_N\) is used if and only if a vehicle carries out \( j \) directly after transport \( i \).
- \( A_K := \{(0, j) : j \in N\} \cup \{(i, n + 1) : i \in N\} \cup \{(0, n + 1)\} \) is the set of arcs from the depot to all \( i \in N \), from each \( j \in N \) to its depot and the arc \((0, n + 1)\) that is used by a vehicle if it does not transport any patients.
- The digraph is \( G := (V, A) \) with \( V := N \cup \{0, n + 1\} \) and \( A := A_N \cup A_K \).
- For each \( i \in N \), \( t_i \) is the target time of transport \( i \).
- For \( k \in K \), \( [a_k, b_k] \subseteq \mathbb{R}_+ \) denotes the shift of the drivers of vehicle \( k \).
- For \( i, j \in N \), \( \tau_{i,j} \geq 0 \) denotes the time a vehicle needs to reach \( O_j \) after starting transport \( i \), i.e., the sum of \( d_i \) and the travel time \( \text{dist}_{i,j} \in \mathbb{R}_+ \) from \( D_i \) to \( O_j \).
For \( j \in N \), \( \tau^k_{0,j} > 0 \) denotes the travel time for vehicle \( k \in K \) to reach \( O_j \) from its depot and \( \tau^k_{j,n+1} > 0 \) denotes the travel time for vehicle \( k \in K \) to reach its depot from \( D_j \). We set \( \tau^k_{0,n+1} := 0 \) for all \( k \in K \).

- \( \gamma_{\text{max}} > 0 \) is the weight for the maximum delay used in the objective.

The variables of our model are

- \( x_{i,j,k} \in \{0,1\} \) is the binary variable that indicates whether vehicle \( k \) travels from node \( i \) to \( j \), i.e., whether the vehicle carries out transport \( j \) directly after it carries out transport \( i \).
- For all \( i \in N \), \( y_i := y_{i,k} \in \mathbb{R}_+ \) denotes the time when a vehicle \( k \) arrives at node \( i \), i.e., when transport \( i \) starts.
- \( y_{0,k} \in \mathbb{R}_+ \) denotes the time when a vehicle starts its trip and \( y_{n+1,k} \in \mathbb{R} \) denotes the time when it ends its trip.

For plannable transports, we minimize the individual delays (possibly weighed by \( \gamma_i \)) and the maximum delay (weighed by \( \gamma_{\text{max}} \)), i.e. we have the (piecewise linear) measure of quality

\[
\gamma_{\text{max}} \cdot \max \{0, y_1 - t_1, \ldots, y_n - t_n\} + \sum_{i \in N} \gamma_i \max \{0, y_i - t_i\}. \tag{4}
\]

Thus, the (linearized) instance of the VRPGTW is Model (5). Here, we introduced the shift times as hard bounds that necessarily need to be satisfied in Constraints (5e) and (5f) since we aim for feasible schedules. In Constraint (5g), we ensure that a vehicle does not start a transport before it is scheduled.

With Model (5), we can consequently establish a feasible schedule: If \( x_{i,j,k} = 1 \), then vehicle \( k \) carries out transport \( i \). Thus, from the optimal solution \( (x^*, y^*) \), it is possible to reconstruct the path of vehicle \( k \) from 0 to \( n + 1 \) of form \((v_{k1}, \ldots, v_{ks})\), where \( s \in \mathbb{N} \) denotes the number of transports for the respective vehicle. Along with the optimal arrival times \( y^* \), an optimal schedule is obtained.

\[
\min \gamma_{\text{max}} \cdot \Delta_{\text{max}} + \sum_{i \in N} \gamma_i \Delta_i, \tag{5a}
\]

\[
\text{s.t.} \quad \text{Constraints (3b) - (3e),} \tag{5b}
\]

\[
y_i + \tau_{i,j} - y_j \leq M_{i,j}(1 - x_{i,j,k}) \quad \forall k \in K, (i, j) \in A_N, \tag{5c}
\]

\[
y_{i,k} + \tau^k_{i,j} - y_{j,k} \leq M_{i,j}(1 - x_{i,j,k}) \quad \forall k \in K, (i, j) \in A_K, \tag{5d}
\]

\[
a_k \leq y_{0,k} \quad \forall k \in K, \tag{5e}
\]

\[
y_{n+1,k} \leq b_k \quad \forall k \in K, \tag{5f}
\]

\[
t_i \leq y_i \quad \forall k \in K, i \in N, \tag{5g}
\]

\[
y_i - t_i \leq \Delta_i \quad \forall i \in N, \tag{5h}
\]

\[
0 \leq \Delta_i \leq \Delta_{\text{max}} \quad \forall i \in N, \tag{5i}
\]

\[
x_{i,j,k} \in \{0,1\} \quad \forall k \in K, (i, j) \in A, \tag{5j}
\]

\[
y_{i,k} \geq 0 \quad \forall i \in N, k \in K. \tag{5k}
\]

We can represent the current penalties for the delays using the penalty weights of Definition 2.8 by

\[
\Gamma := \{(1, \{\Delta_i \mid i \in N\}) \cup \{(\gamma_{\text{max}}, \{\Delta_{\text{max}}\})\},
\]

so we do not state the delays explicitly in the following.

**Additional Requirements** Besides the extensions for transports, in the following we are going to present different modeling possibilities that allow us to extend Model (5) in various ways. In Subsection 3.4, we use these techniques to model the difficulties and additional requirements during the Covid-19 pandemic.
One possibility is to limit the number of transports per vehicle with a label $\psi \in \Psi$ to a fixed number $n \in \mathbb{N}$. This can be modeled by

$$\sum_{(i,j) \in A} x_{i,j,k} \cdot \psi(i) \leq n \ \forall k \in K.$$  

(6)

If a specific label $\psi$ is not of importance, then we can just omit it. With $n = 1$, each vehicle can handle at most one transport of a specific type. Let $\psi_1, \psi_2 \in \Psi$ and we assume that it is not allowed to handle a transport $j$ with $\psi_2(j) = 1$ immediately after a transport $i$ with $\psi_1(i) = 1$. This is represented by adding constraints

$$\psi_1(i) \cdot \psi_2(j) \cdot x_{i,j,k} = 0 \ \forall k \in K, (i,j) \in A.$$  

(7)

to Model (5). Of course, it is possible to use $\psi_1 = \psi_2$ and, thus, prohibit the handling of similar transports after another.

### 3.3 Incorporating of transports with incomplete information

For incorporating semiplannable transports in our model, we apply the strategy for the operational phase that has been proposed in Section 2: Whenever possible, we estimate the target times of semiplannable transports and treat dummy transports like plannable, i.e., the VRPGTW instance Model (5) is modeled with plannable transports and dummy transports. To take the data–driven nature of dummy transports into account, we establish the following rules: Firstly, if the maximum delay is a measure of quality, delays of dummy transports are not taken into account (i.e., only the delays of the plannable transports are relevant). Secondly, the individual delay of dummy transports is a quality of measure and is weighted by some parameter $\gamma_i < 1$ for $i \in T_{\text{dummy}}$. This can be further evaluated, but at the moment it is only important that the weight is smaller than 1 as we use a value of 1 for plannable transports. Finally, if its actual target time is higher than its waiting time, the quality of the schedule decreases. Thus, after incorporating the dummy transports, the objective of Model (5) is

$$\gamma_{\text{max}} \cdot \Delta_{\text{max}} + \sum_{i \in N} \gamma_i \Delta_i + \sum_{i \in T_{\text{dummy}}} \gamma_i \max\{0, y_i - t_i^{\text{est}}\} + M_i \max\{0, y_i - (t_i + w_i)\}$$  

(8)

with $M_i \in \mathbb{R}_{+}$, $i \in T_{\text{dummy}}$, being a sufficiently large number. With the scheduling of all transports in $T_{\text{plan}}$, the planning phase is completed. In addition to semiplannable transports, the dispatchers usually have to incorporate ad–hoc transports. They are called whenever such a transport has to be scheduled at a certain time $\sigma$ on the fly. Since at this point of time a schedule is already carried out, some vehicles are already on their trips. Furthermore, serving this ad–hoc transport using a vehicle in the depot without respecting schedules can cause long delays for already planned transports.

We now introduce some notation. Let $\sigma \in \mathbb{R}_{+}$ be a given point of time. The set $V^{\sigma} \subseteq V$ denotes all nodes that are known before the new ad–hoc transport is requested. The set

$$\mathcal{P}^{\sigma} := \{(i,j,k,y_i^*) \in V^{\sigma} \times V^{\sigma} \times K \times \mathbb{R}_{+} : x_{i,j,k}^* = 1\}$$

denotes the schedule that is carried out at $\sigma$. If no ad–hoc transports have been requested before $\sigma$, we have $\mathcal{P}^{\sigma} = \mathcal{P}^{\emptyset}$. Transports already in operation are not changed. They are elements of the set

$$\mathcal{P}_{\text{fixed}}^{\sigma} := \{(i,j,k,y_i^*) \in \mathcal{P}^{\sigma} : \sigma \geq y_{j,k} - \text{dist}_{i,j}\}.$$  

that contains all transports, where the vehicle is either on the way from $D_i$ to $O_j$ or has even started or finished transport $j$. If an ad–hoc transport is requested for some time point $\sigma$, transports of $\mathcal{P}_{\text{fixed}}^{\sigma}$ are not changed while the transports in $\mathcal{P}^{\sigma} \setminus \mathcal{P}_{\text{fixed}}^{\sigma}$ are removed of the schedule. This ensures that the vehicles that have not been transporting patients are available again since their schedule after $\sigma$ was deleted. Thus, every transport scheduled after $\sigma$, including the ad–hoc transport, can be re–scheduled by solving Model (5) with the additional restriction that all transports in $\mathcal{P}_{\text{fixed}}^{\sigma}$ are unchanged. Therefore, we first update $G$ by including the new ad–hoc transport in $V^{\sigma}$ and
\[ A' := \{(i, j) \in A : i, j \in V'\} \], respectively. Thereupon, to ensure that fixed transports are not changed, we introduce two additional constraints, namely

\[ x_{i,j,k} = x^*_{i,j,k} \forall (i, j, k, y^*) \in P^*_{\text{fixed}} \]  
(9)

and

\[ y_j = y^*_j \forall (i, j, k, y^*) \in P^*_{\text{fixed}} \].  
(10)

### 3.4 Adjusting the model during the Covid–19 pandemic

To decrease the risk of infections for patient transports during the Covid–19 pandemic, it is desirable that transports are distinguished into two types, depending on whether a patient is (suspected to be) infected with Covid–19 or not - resulting in an additional property of transports. From now on, with 'infected', we mean a (suspected) infection with Covid–19. In the following, we will discuss some ideas how such Covid–19 transports can be handled and proceed with a mathematical formulation how minimizing the risk of infections can be incorporated in our Model (5).

#### 3.4.1 Approaches to handle Covid–19 transports

There are different possibilities to decrease the risk of infections. This has been investigated in e.g. [25]. In practice, when dealing with dangerous infectious illnesses like Covid–19, the staff has to wear protective clothing and the vehicle is disinfected after every transport. In addition to these protective measures, it can be helpful to minimize the number of changes from a Covid–19 transport to a non–Covid–19 transport. This will reduce the number of contacts between patients and staff and thus the risk of infection even further.

We present two different approaches, namely reducing the number of vehicles that are allowed to carry infected patients and limiting the number of infected patients per vehicle. Additionally to the aim of reducing the number of changes, these approaches have been developed due to the fact that the supply of protective clothing is limited and therefore it should be distributed as efficiently as possible. In both cases, we still aim to minimize the waiting times for patients. This is handled by using different penalty parameters in the objective function.

To indicate, for which transports additional Covid–19 requirements are necessary, we introduce a label \( c \): \( N \times \{0, n + 1\} \rightarrow \{0, 1\} \). A value of 1 means that a patient is infected. For the depot, i.e., \( i \in \{0, n + 1\} \), we have \( c(i) = 0 \). This label is used to create additional constraints and we write \( c_i \) instead of \( c_{i,j} \). Furthermore, we write \( \bar{c} \), meaning a transport is not a Covid–19 transport, with \( \bar{c}_i = 1 - c_i \) for all \( i \in N \cup \{0, n + 1\} \).

**Minimizing the number of changes** We can use the labels \( c \) and \( \bar{c} \) with Constraint (7):

\[
\bar{c}_i \cdot \bar{c}_j \cdot x_{i,j,k} = 0 \forall k \in K, \ (i, j) \in AN
\]  
(11)

This equation models that no vehicle is allowed to carry a Non-Covid–19 transport directly after a Covid–19 transport. We do not want to prohibit all changes because otherwise, the solution quality will decrease or we are not able to create feasible schedules anymore. Therefore, we use a soft constraint instead of minimizing the number of changes and we create the additional variable \( \lambda^\text{change}_{i,j,k} \in \{0, 1\} \) for \( (i, j) \in AN, k \in K \) and modify Constraint (11) to:

\[
\bar{c}_i \cdot \bar{c}_j \cdot x_{i,j,k} = \lambda^\text{change}_{i,j,k} \forall k \in K, \ (i, j) \in AN.
\]  
(12)

Now the variables \( \lambda^\text{change}_{i,j,k} \) are penalized using \( \gamma^\text{change} \) and we modify \( \Gamma \) by appending the tuple \( (\gamma^\text{change}, \{\lambda^\text{change}_{i,j,k} \mid (i, j) \in AN, k \in K\}) \).

**Additional time effort** The additional time for disinfection and changing of clothes must be taken into account when establishing the schedules. This is easily done by increasing the duration \( d_i \) of each transport by a fixed value. Thus, it is not necessary to create additional constraints.
Minimizing the number of Covid–19 vehicles

An option to reduce contact between uninfected and infected person is dividing the vehicle fleet into different pools, i.e. one pool of vehicles that only handle Covid–19 transport and one pool for all other forms of transport. It is possible to additionally use so–called floater vehicles that are allowed to handle both types of transports to maintain some degree of flexibility. Here, the protective clothing can be distributed among the Covid–19 vehicles (and the floater vehicles) so that no vehicle is carrying it needlessly.

The label $c_i$ is used to indicate whether a vehicle is considered as a Covid–19 vehicle. The corresponding constraint is

$$c_i \cdot x_{i,j,k} \leq \lambda^\text{vehicle}_k, \forall k \in K, (i,j) \in A.$$  

Here, $\lambda^\text{vehicle}_k$ for $k \in K$ are new binary variables. They are penalized using $\gamma^\text{vehicle}_k$, i.e., we update $\Gamma$: $\Gamma := \Gamma \cup \{\gamma^\text{vehicle}_k, \{\lambda^\text{vehicle}_k \mid k \in K\}\}$

Limiting the number of Covid–19 transports for each vehicle

A different approach is distributing the protective clothing uniformly between all vehicles. In this case, every vehicle is able to serve a limited number of Covid–19 transports before it has to return to its depot to obtain a new set of clothing. An advantage of this approach is that it takes into account the fact that each vehicle is able to handle Covid–19 transports and so some of them may be handled with a smaller delay than when separating the fleets completely. Nevertheless, this modeling approach can result in an increased number of switches between infected and non–infected patients for each vehicle.

To implement the limitation of protective clothing, we use a tuple of a penalty weight and integer variables, namely $(\gamma^\text{clothing}_k, \Lambda^\text{clothing}_k)$ where $\Lambda^\text{clothing}_k := \{\lambda^\text{clothing}_k \mid k \in K\}$. The penalty is applied every time a vehicle has to return to the depot to obtain new clothing. Thus, we introduce the constraints

$$\sum_{(i,j) \in A} c_i \cdot x_{i,j,k} \leq \alpha + \lambda^\text{clothing}_k, \forall k \in K$$  

where $\alpha$ is the number of clothings per vehicle. This is Constraint (6), with the difference that not every exceedance of $\alpha$ is penalized. Instead, every time $\alpha$ is reached again, we increment the variable $\lambda^\text{clothing}_k$ by one. In our model, it is not possible that vehicles return to the depot during their shift. Thus, the penalty for this scenario is chosen quite high so a vehicle only exceeds its limit when it is not avoidable, i.e. if there are more than $\alpha \mid K \mid$ Covid–19 transports.

This concludes the modeling of the patient transport problem. In the next section, we present and discuss our numerical experiment and show the efficiency of our approach.

4 Implementation and numerical results

In this section, we provide details on the implementation as well as the results that can be obtained from the optimized schedules. The models presented in the earlier section are solved via state–of–the–art available global MIP solvers like Gurobi, cf. [20], [30] or [31]. The rationale behind the usage of available software is to enable possible transfer of the developed approaches to the practitioners without any further modifications using combinatorial algorithms. We start with the description of a simulated reality to evaluate our schedules in Subsection 4.1. Thereupon we present some heuristic approaches in Subsection 4.2 and, finally, we evaluate the performance of our approach in Subsection 4.3. Therein we also cover the incorporation of semiplannable transports and the extensions to cover the Covid–19 pandemic.

All models and algorithms were implemented in Python 3.7.7. To solve the resulting MIPs, we used Gurobi 9.0.2 [17] on machines with Intel Xeon E3-1240 v5 or Intel Xeon E3-1240 v6 CPUs, respectively. Each of these has four cores with 3.5 GHz each and a RAM of 32 GB.

For all numerical results, we use historical data from Middle Franconia. In practice, this area is subdivided into different counties, and a separate scheduling is made for each of them whereby the transports $T$ are assigned depending on their origin location $O_T$. We proceed analogously the same with our optimization approach. These areas are of different size, and have a different number of inhabitants. We have rural areas with a small number of inhabitants compared to the size as well as urban areas with a large population density. We optimize each day separately because
they do not influence each other as during the night almost no transports are requested. Unless otherwise mentioned, we specify a time limit of 60 minutes (we will elaborate on the time limit in Subsection 4.3).

The available historical data have some problems we have to overcome: Firstly, some transports are stated incorrectly. For example some timestamps or locations are missing. A second point relates to practical short-term decisions made by the dispatcher. One example is that vehicles can be lent between different counties. Technically, each vehicle is assigned to one area but in exceptional situations, this can be softened. The same holds for the distinction of patient and rescue transports that can be disregarded when absolutely necessary. Further, it is possible to postpone a transport to another day if the delay would increase too much.

During preprocessing, the problem of missing data was handled by either estimating the missing data or by deleting the corresponding transports. Furthermore, we do not consider human decisions in our model since we are not in a position to make them. In practice, our schedules after optimization can possibly be improved by incorporating expert decisions.

4.1 Implementation of a simulated reality

Due to the reasons described beforehand, we cannot compare our schedules directly to historical ones. So, in order to evaluate the optimized schedules, we require a baseline solution. To have the best possible comparison, we implement a simulation for the decisions made by the ILS that works similar to the practice. They use a greedy approach: Every time a transport is necessary, the vehicle that can arrive the fastest is assigned to this task. However, the availability of vehicles must be carefully considered, i.e., vehicles currently involved in transportation cannot be used and shift times are respected whenever possible.

For the baseline implementation, we thus sort all transports by their target time in the historical data. Then the vehicles are assigned in that order: For each vehicle, we calculate the earliest time it could reach the origin of the requested transport by adding the travel time to the time it is expected to become available, which is either the start of its shift or the expected end of the previous transport. Furthermore, we check whether the vehicle could reach its depot without violating its shift times using the estimated duration of the transport.

The vehicle that can carry out the transport request with the smallest delay is assigned to it. If there is more than one vehicle with minimal delay, then the one with the shortest travel time is chosen. In the case that no vehicle can handle the transport without violating its shift times, we choose the vehicle with the smallest violation time.

4.2 Heuristic methods for the MIP solver

We present some algorithmic ideas to speed up the MIP solution process for the models from the earlier sections, because a solution approach without any improvements is not able to solve many instances to optimality. From the real historical data, we use a subset, namely the data corresponding to 59 days (January and February 2020) from two areas, i.e. 118 instances in total. The areas have different sizes, the smaller area has about 100,000 inhabitants, the larger area about 500,000.

Using a primal heuristic The first heuristic we implement is a primal heuristic that aims to find good feasible solutions early in the MIP solution process. At every $k$-th node of the branch-and-bound tree within the MIP solver, we tentatively round a part of the optimal solution of the LP relaxation. In every feasible solution, the number of variables $x_{i,j,k}$ that are set to 1 is given by $|K| + |V|$. For a primal heuristic, all values for $x_{i,j,k}$ in the solution of the LP relaxation are sorted and a particular number of them are fixed in order to faster obtain a feasible mixed-integer problem (MIP) solution. Here, we have chosen the number of vehicles $|K|$ and set the $|K|$ greatest variables to 1. If a feasible solution is found by using this start solution, then it is used for the following solving process. Otherwise, it is discarded. It is important not to fix too many variables as the solution may then become infeasible. In contrast, fixing too few variables will not speed up the solving process. All these values are based on empirical analysis.
Table 1: Comparison of different speed up methods for smaller problems in the upper table and larger problems in the lower table. The unit s stands for seconds.

<table>
<thead>
<tr>
<th></th>
<th>Primal heuristic</th>
<th>Greedy heuristic</th>
<th>No heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average time for solved problems</td>
<td>69.3s</td>
<td>58.0s</td>
<td>62.8s</td>
</tr>
<tr>
<td>Solved problems in 60 minutes</td>
<td>55</td>
<td>55</td>
<td>55</td>
</tr>
<tr>
<td>Solved problems in 10 minutes</td>
<td>54</td>
<td>55</td>
<td>55</td>
</tr>
<tr>
<td>Solved problems in 5 minutes</td>
<td>53</td>
<td>55</td>
<td>55</td>
</tr>
<tr>
<td>Solved problems in 3 minutes</td>
<td>53</td>
<td>55</td>
<td>53</td>
</tr>
<tr>
<td>Average time for solved problems</td>
<td>520.3s</td>
<td>437.8s</td>
<td>642.6s</td>
</tr>
<tr>
<td>Solved problems in 60 minutes</td>
<td>27</td>
<td>33</td>
<td>31</td>
</tr>
<tr>
<td>Solved problems in 30 minutes</td>
<td>23</td>
<td>32</td>
<td>26</td>
</tr>
<tr>
<td>Solved problems in 15 minutes</td>
<td>21</td>
<td>27</td>
<td>22</td>
</tr>
<tr>
<td>Solved problems in 10 minutes</td>
<td>21</td>
<td>25</td>
<td>22</td>
</tr>
</tbody>
</table>

Using a greedy heuristic to obtain a first feasible solution The simulation of the current scheduling approach given in Subsection 4.1 creates an (at least weakly) feasible schedule. This schedule, that can be computed very quickly, can be used as a starting solution by constructing a solution for the variables \(x_{i,j,k}\) – the remaining variables depend on the values for \(x_{i,j,k}\) and, thus, are directly calculated by the solver. This greedy heuristic can be applied once at the beginning of each (re-)scheduling process.

Determining a branching priority Within each branching step, a variable is chosen from all binary and integer variables, depending on some measure that aims to create preferably small branch-and-bound trees, see for example [1]. Let us consider the determination of optimized plans under Covid-19 restrictions as described in Subsection 3.4. If the values for the binary variables \(x_{i,j,k}\) are decided, the unique values for the remaining binary and integer variables can easily be derived. The values for the remaining binary variables, i.e. Covid-19 vehicle and change labels, are directly implied, see Constraints (12), (13) and (14). The values for \(y_i\) represent the time when a patient is picked up. Starting at the depot, the values are calculated successively. This is possible because each value only depends on its unique predecessor, so we can simply follow the paths of the vehicles. Thus, we obtain a branching priority and branch on the former variables first that we provide as input.

Handling of difficult instances Especially on days where many transports have to be scheduled, the solving process of the resulting MIPs can take a long time. We have days with around 140 transports in our data, as presented in Section 1. While we can limit the total running time, we can also directly limit the time required for each MIP. To accomplish this, the solving process is interrupted when the time limit is reached and the current best solution is used for establishing a schedule at this time. To ensure that a (weakly) feasible solution is found, we provide a start solution, namely the previously described solution of our greedy heuristic. The idea behind this is that short-term decisions, i.e., those that must be made in the near future, can be dealt with directly while long-term decisions can be rescheduled at a later stage. Further possibilities for improving the VRPGTW’s solving process, e.g. determining bounds of the optimal solution, are discussed in [9].

Next, we evaluate the performance of our approach. Therein we also cover the incorporation of semiplannable transports and the extensions to cover the Covid-19 pandemic.

Comparison of the heuristic methods In Table 1, both heuristic methods to enhance the MIP solver are compared to each other and to the results when no such heuristic is used. We evaluate on both an easier set and a more difficult set with larger instances. In the easier test set around 20 transports per day are requested, while in the more difficult test set there are up to 65 transports per day. During the weekend, the instances are smaller. The number of available vehicles is between 5 and 20, depending on the weekday and the size of the considered county.
The number of binary and continuous variables are estimated by $\mathcal{O}(|K| \cdot |V|^2)$ and $\mathcal{O}(|K| \cdot |V|)$, respectively. The number of constraints is also quadratic in the number of patients, more precisely it is given by $\mathcal{O}(|K| \cdot |V|^2)$. In general, the number of transports is greater than the number of vehicles. On the easier problem set, as shown in the upper part of Table 1, all approaches are able to solve most of the problems to optimality. The application of a greedy heuristic is slightly faster while a primal heuristic is slightly slower but there are no significant differences.

In contrast, on the more difficult test set, the usage of the greedy heuristic yields a high improvement. Compared to the other approaches, problems are solved significantly faster. If we use the baseline solution, then a weakly feasible solution is given at the root node, where many branch-and-bound nodes that yield a worse solution can be pruned. In fact, in most cases, the baseline solution is even feasible and not only weakly feasible, which further improves the run time of our algorithm. The solution time is improved as a result of the branching priority from Subsection 4.2. This enables us to increase the number of problems solved within 60 seconds by around 50%. Note that if we do not consider Covid–19 requirements, this priority will have no impact. Thus, we always prioritize the variables $x_{i,j,k}$ for branching.

### 4.3 Improvements using the VRPGTW model

For the following results, we applied the greedy heuristic together with a branching priority. The proposed results are chosen exemplarily and are characteristically similar for other areas. In the following, we compare the historical data’s solution and the existing course of action with our optimization approach. Firstly, we consider the results of our general model without any extensions in Subsubsection 4.3.1. Secondly, in Subsubsection 4.3.2, we consider insights that can be drawn from these results. Finally, in Subsubsection 4.3.3 and Subsubsection 4.3.4 we present the results for our extensions for semiplannable transports and the Covid–19 pandemic.

#### 4.3.1 Planning of plannable and ad–hoc transports without further requirements

Now, we evaluate the optimized schedules against the baseline solution of Subsection 4.1. We use the data from 2019 as there have been no Covid–19 transports that distort the scheduling process. Unless stated otherwise, semiplannable transports are treated as ad–hoc transports. As an example, we show the delays for different regions. We start by comparing the largest delay of the baseline schedule and our optimization procedure for all instances that could be solved within 60 minutes. Instances that are not solved until optimality at this time are very likely also not solvable in a significant higher time limit.

In Figure 3, they are compared for two different regions, Erlangen at the top and the county Erlangen–Höchstadt at the bottom. In each plot, the difference between the maximum delays, computed by the optimization model and by the baseline, is shown. A positive value means that an improvement could be achieved, while a negative value means that the maximum delay has increased compared to the present scheduling approach. Furthermore, days for which a reduction in shift time violations could be achieved are highlighted. This is shown by the red bars.

In the upper plot, an improvement with regard to shift times was achieved in nine days. As respecting shift times is highly desirable, a somewhat longer maximum delay is acceptable. Apart from that, the optimized maximum delay is up to 60 minutes shorter than that of the baseline solution. In particular, there is one day for which the improvement is much larger than on the other days. On this day, many transports were requested for the late afternoon, which shows results for problems in a larger but more rural area. In such an area, there are less available vehicles but also a smaller number of requested transports. The optimized schedule often decreases the maximum delay by up to 60 minutes. Furthermore, on two days, there is a reduced number of shift time violations as well as a smaller maximum delay. The overall delay is reduced by about 10 minutes on a daily average.

There are also instances where optimized plans perform worse than the baseline scheduling: On one day, the optimized schedule yields a maximum delay that is around ten minutes longer than that of the baseline schedule. This is caused by some ad–hoc transports. The schedule of the optimized

---

2We recall that we respect shift times in our model and only violate them if otherwise no feasible schedule could be obtained.
Figure 3: Improvement in maximum delay that can be gained using our optimization approach, when compared to the simulated reality. If a difference is highlighted in red, then this means that an improvement in shift time violations could be achieved. Two different regions are considered, the above one is a smaller urban region while the bottom one is larger but more rural.

solution up to this point might be at least as good as the baseline schedule but the vehicles are organized differently. Then, although the baseline schedule is not necessarily the optimum of Model (5), it could lead to a better complete schedule when a new request becomes known. This is not completely preventable as there is no information about such transports earlier in the day. However, our approach from Subsection 3.3 aims to prevent such settings by including dummy transports, and so this situation occurs only rarely.

Using a smaller time limit for each MIP Now, instead of solving each MIP of an instance to optimality and aborting after a total time limit, we specify a time limit after which the solving process of a single MIP is aborted and the current best, w.r.t to the objective of Model (5), is used. For practical reasons, there is still a total time limit of 24 hours, but this was never reached in our experiments. Using this approach, it is possible to solve problems that occur in larger areas, i.e., such with up to 130 transports per day, with more vehicles and transports. In the following, we evaluate the resulting schedules. Figure 4 shows a comparison to the simulated reality described in Subsection 4.1. The representation is similar to that in Figure 3. Here, we specified a time limit of 10 minutes per MIP. We figured out that there is only a small difference between 10 and 60 minutes, thus, we use the smaller one as this is closer to reality. In many instances there is a much greater delay, which results from the fact that we are able to decrease the number and duration of shift time violations. Sometimes, it is even possible to find a feasible solution where the baseline solution is just weakly feasible – so it is possible to respect the drivers’ shift times. For the remaining problems, we decrease the maximum delay by up to 32
minutes, with an average improvement of around 5 minutes. In smaller areas this value is much higher.

The remaining question centers on how good the resulting schedules are compared to the optimal solution. To determine this, we can evaluate how many problems could be solved to optimality. For example, in 289 of 364 instances, we found an optimal solution for at least 80% of all MIPs. An optimal solution for every MIP could be found in 267 instances.

Many of the problems where we are not able to compute a good solution are (presumably) only weakly feasible. Here there are two goals, namely minimizing the delay as well as the shift time violations, which increases the difficulty of the problem.

4.3.2 Insights from the optimal solution

In Subsubsection 4.3.1, we have evaluated the optimization results and discovered that they improve the current course of action.

We examine how the delays vary over the course of the day. In Figure 5a, the maximum and average delay, and in Figure 5b, the total delay, are differentiated by target time.

Figure 5: Delays in the optimal schedules for instances of 2019, differentiated by target time.

In Subsubsection 4.3.1, we have evaluated the optimization results and discovered that they improve the current course of action.

We examine how the delays vary over the course of the day. In Figure 5a, the maximum and average delay, and in Figure 5b, the total delay, are differentiated by target time.
average delay aggregated over the year is plotted for every hour. The average delay is calculated using the number of transports. In Figure 5b, we further give the total delay over the year. The maximum delay is roughly equal throughout the day. From the late afternoon, there is nearly no delay, with one exception at 5pm. This is presumably caused by shift ends. The average delay is highest in the early morning and decreases continuously. To gain a better understanding of what this means, we also consider the total delay on the right hand side. Here, we have high delays from 6am, not from 5am as for the maximum and average delay. While only a small number of transports are needed at this time, there are only few vehicles with a shift at this time, so the average waiting time for these patients becomes very high. This continues with the transports starting shortly after 6am. As the morning shift times do not start earlier than 6am and they first need to drive to the origin of a transport, it is not possible to reach these patients without a delay. In general, the night shifts end at 6am, and, thus, they can probably not handle these transport requests without risking shift time violations.

A more detailed evaluation shows that the vehicles with earlier shift times have to handle considerably more transports than the later ones. Vehicles are working almost at full capacity at least until noon. Afterwards, less ad–hoc transports are requested and the situation eases. This means that as soon as a vehicle becomes available, it will be assigned to some transport. As few vehicles are available at the beginning of a day, some transports cannot be handled on time. These delays cause delays for later transports. This occurs because vehicles are occupied by earlier transports that are preferred as we minimize the maximum delay. In the afternoon, less transports are requested. Thus, more vehicles are available and the delays finally decrease.

4.3.3 Incorporating semi-plannable transports

In this subsubsection, we cover the usage of dummy transports for semi-plannable transports. In practice, there are some problems with the provided data. At the beginning of the optimization, each transport \( T \) is assigned to an area depending on the county of their origin location \( O_T \). Often, the destination \( D_T \) is located in another area. In this case, patients have to be transported between different areas, causing outward and return journeys not being represented in the same model. This leads to dummy transports that are not applied in the actual return journey. In practice, the dispatcher can assign such transports to the same area. Furthermore, in newer data, a type of dummy node is used. As soon as a dialysis is requested, two transports are created, including the return trip on 23:59 on the same day. Its target time is corrected as soon as it becomes known. Thus, if a dialysis is created at least one day before, then it is a plannable transport but its return trip is not, although it is stated to be plannable in the data. Now, we consider a schedule where dummy transports actually decrease the total and maximum delays. Table 2a and Table 2b display two schedules that were created on an example day. The first is the current schedule shortly before 11:00. At 11:00, a new transport is requested. In fact, this transport is a return journey for a previous dialysis transport for which a dummy node has been created. The outward journey has ID 28 and thus the dummy node is called \( d28 \). It is recognized that the new transport with ID 20 corresponds to this dummy node and thus, at this point, it is deleted and replaced by the new transport using the new information, in particular the actual schedule time. Table 2b shows the schedule created at 11:00. As can be seen, the vehicles allocated to the transports not contained in \( P_{\text{fixed}}^\sigma \) for \( \sigma = 11:00 \), have changed. This is due to the fact that the transport is requested a little later than presumed and vehicle 7 is available at this point. The schedule shows that it finishes its previous transport at 11:48, the new transport is requested at 12:00 instead of the presumed 11:52. So, this vehicle can reach the origin of the return journey in time. Compared to the schedule that is created without dummy nodes, the maximum delay decreases by 20 minutes. Thus, the total delay also decreases. In this case, we create only one dummy node that is replaced later on. Taking this into consideration, this is a very positive result and an even higher improvement can be expected if more dummies could be used whenever appropriate data is available.
Table 2: A schedule before and after transport 20 has been replaced by its dummy node d28. The second dummy node, d24 for transport 24, does not correspond to a future transport and will thus not be replaced before the end of the day. The information about the return trip became known at 11:00, all fixed transports are given below the line, so the remaining ones could be rescheduled. As can be seen, the delays remain the same for all transports. Note that there are relatively few transports in the afternoon as they are often ad-hoc ones and thus are not known at this point.

(a) Schedule before 11:00.

<table>
<thead>
<tr>
<th>ID (i)</th>
<th>t_i</th>
<th>k</th>
<th>start (y_i)</th>
<th>end</th>
<th>∆t</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>06:45</td>
<td>0</td>
<td>06:45</td>
<td>07:52</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>07:30</td>
<td>13</td>
<td>07:30</td>
<td>11:57</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>08:00</td>
<td>0</td>
<td>08:00</td>
<td>08:33</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>08:00</td>
<td>6</td>
<td>08:03</td>
<td>11:01</td>
<td>3.6529</td>
</tr>
<tr>
<td>25</td>
<td>08:30</td>
<td>0</td>
<td>08:33</td>
<td>10:00</td>
<td>3.2262</td>
</tr>
<tr>
<td>11</td>
<td>08:45</td>
<td>7</td>
<td>08:45</td>
<td>09:46</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>09:00</td>
<td>5</td>
<td>09:40</td>
<td>10:02</td>
<td>4.2565</td>
</tr>
<tr>
<td>0</td>
<td>09:00</td>
<td>8</td>
<td>09:40</td>
<td>11:10</td>
<td>4.2565</td>
</tr>
<tr>
<td>26</td>
<td>09:45</td>
<td>10</td>
<td>09:45</td>
<td>11:01</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>09:45</td>
<td>7</td>
<td>09:47</td>
<td>10:37</td>
<td>2.0320</td>
</tr>
<tr>
<td>13</td>
<td>10:00</td>
<td>10</td>
<td>10:45</td>
<td>11:59</td>
<td>45.0436</td>
</tr>
<tr>
<td>3</td>
<td>10:00</td>
<td>1</td>
<td>10:33</td>
<td>11:46</td>
<td>33.8629</td>
</tr>
<tr>
<td>12</td>
<td>10:00</td>
<td>0</td>
<td>10:00</td>
<td>10:58</td>
<td>0.5156</td>
</tr>
<tr>
<td>14</td>
<td>10:00</td>
<td>5</td>
<td>10:14</td>
<td>10:56</td>
<td>14.6932</td>
</tr>
<tr>
<td>15</td>
<td>10:00</td>
<td>7</td>
<td>10:37</td>
<td>11:48</td>
<td>37.1338</td>
</tr>
<tr>
<td>21</td>
<td>10:00</td>
<td>10</td>
<td>10:05</td>
<td>10:43</td>
<td>4.8170</td>
</tr>
<tr>
<td>7</td>
<td>10:15</td>
<td>5</td>
<td>10:59</td>
<td>12:18</td>
<td>44.0412</td>
</tr>
<tr>
<td>16</td>
<td>10:30</td>
<td>0</td>
<td>10:58</td>
<td>11:44</td>
<td>27.8051</td>
</tr>
<tr>
<td>9</td>
<td>10:30</td>
<td>11</td>
<td>11:03</td>
<td>12:30</td>
<td>33.8629</td>
</tr>
<tr>
<td>18</td>
<td>10:40</td>
<td>10</td>
<td>11:23</td>
<td>12:22</td>
<td>44.1832</td>
</tr>
<tr>
<td>1</td>
<td>11:00</td>
<td>0</td>
<td>11:44</td>
<td>12:31</td>
<td>44.0945</td>
</tr>
<tr>
<td>d28</td>
<td>11:52</td>
<td>1</td>
<td>11:52</td>
<td>12:59</td>
<td>0</td>
</tr>
<tr>
<td>d24</td>
<td>12:33</td>
<td>10</td>
<td>12:33</td>
<td>13:08</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>15:30</td>
<td>2</td>
<td>15:30</td>
<td>18:21</td>
<td>0</td>
</tr>
</tbody>
</table>

(b) Schedule at 11:00.

<table>
<thead>
<tr>
<th>ID (i)</th>
<th>t_i</th>
<th>k</th>
<th>start (y_i)</th>
<th>end</th>
<th>∆t</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>06:45</td>
<td>0</td>
<td>06:45</td>
<td>07:52</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>07:30</td>
<td>13</td>
<td>07:30</td>
<td>11:57</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>08:00</td>
<td>0</td>
<td>08:00</td>
<td>08:33</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>08:00</td>
<td>6</td>
<td>08:03</td>
<td>11:01</td>
<td>3.6529</td>
</tr>
<tr>
<td>25</td>
<td>08:30</td>
<td>0</td>
<td>08:33</td>
<td>10:00</td>
<td>3.2262</td>
</tr>
<tr>
<td>11</td>
<td>08:45</td>
<td>7</td>
<td>08:45</td>
<td>09:46</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>09:00</td>
<td>5</td>
<td>09:40</td>
<td>10:02</td>
<td>4.2565</td>
</tr>
<tr>
<td>0</td>
<td>09:00</td>
<td>8</td>
<td>09:40</td>
<td>11:10</td>
<td>4.2565</td>
</tr>
<tr>
<td>26</td>
<td>09:45</td>
<td>10</td>
<td>09:45</td>
<td>11:01</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>09:45</td>
<td>7</td>
<td>09:47</td>
<td>10:37</td>
<td>2.0320</td>
</tr>
<tr>
<td>13</td>
<td>10:00</td>
<td>10</td>
<td>10:45</td>
<td>11:59</td>
<td>45.0436</td>
</tr>
<tr>
<td>3</td>
<td>10:00</td>
<td>1</td>
<td>10:33</td>
<td>11:46</td>
<td>33.8629</td>
</tr>
<tr>
<td>12</td>
<td>10:00</td>
<td>0</td>
<td>10:00</td>
<td>10:58</td>
<td>0.5156</td>
</tr>
<tr>
<td>14</td>
<td>10:00</td>
<td>5</td>
<td>10:14</td>
<td>10:56</td>
<td>14.6932</td>
</tr>
<tr>
<td>15</td>
<td>10:00</td>
<td>7</td>
<td>10:37</td>
<td>11:48</td>
<td>37.1338</td>
</tr>
<tr>
<td>21</td>
<td>10:00</td>
<td>10</td>
<td>10:05</td>
<td>10:43</td>
<td>4.8170</td>
</tr>
<tr>
<td>7</td>
<td>10:15</td>
<td>5</td>
<td>10:59</td>
<td>12:18</td>
<td>44.0412</td>
</tr>
<tr>
<td>16</td>
<td>10:30</td>
<td>0</td>
<td>10:58</td>
<td>11:44</td>
<td>27.8051</td>
</tr>
<tr>
<td>9</td>
<td>10:30</td>
<td>11</td>
<td>11:03</td>
<td>12:30</td>
<td>33.8629</td>
</tr>
<tr>
<td>18</td>
<td>10:40</td>
<td>10</td>
<td>11:23</td>
<td>12:22</td>
<td>44.1832</td>
</tr>
<tr>
<td>1</td>
<td>11:00</td>
<td>0</td>
<td>11:44</td>
<td>12:31</td>
<td>44.0945</td>
</tr>
<tr>
<td>d28</td>
<td>11:52</td>
<td>1</td>
<td>11:52</td>
<td>12:59</td>
<td>0</td>
</tr>
<tr>
<td>d24</td>
<td>12:33</td>
<td>10</td>
<td>12:33</td>
<td>13:08</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>15:30</td>
<td>2</td>
<td>15:30</td>
<td>18:21</td>
<td>0</td>
</tr>
</tbody>
</table>
4.3.4 Handling of Covid–19 requirements

For our numerical study, we assume that the dispatcher does not have a special course of action for the Covid–19 pandemic, i.e., it treats transport for infected patients in the same way as other transport. Thus, it is not helpful to evaluate our model from Subsection 3.4 against the (simulated) reality as we did before, because an explicit minimization of infection risks is not considered there. Instead, we aim to obtain new insights into how the pandemic requirements could be handled using a similar approach to that in Subsubsection 4.3.2. This is accomplished for the approach assigning the vehicles into pools. In particular, we evaluate this pool division and how it can be used in practice.

Table 3: Optimized pools for vehicles in one example area and the example shift times for Tuesdays. For each shift time, the number of handled transports, as well as the percentage of the allocation to each pool are given.

<table>
<thead>
<tr>
<th>Shift time</th>
<th>Number of vehicles</th>
<th>Number of transports</th>
<th>Covid–19 vehicles</th>
<th>Non-Covid vehicles</th>
<th>Floater vehicles</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00 - 14:00</td>
<td>1</td>
<td>189</td>
<td>2.9%</td>
<td>94.2%</td>
<td>2.9%</td>
</tr>
<tr>
<td>07:00 - 15:00</td>
<td>1</td>
<td>191</td>
<td>1.5%</td>
<td>91.2%</td>
<td>7.3%</td>
</tr>
<tr>
<td>08:00 - 16:00</td>
<td>1</td>
<td>161</td>
<td>2.9%</td>
<td>90.0%</td>
<td>7.1%</td>
</tr>
<tr>
<td>08:30 - 16:30</td>
<td>1</td>
<td>135</td>
<td>1.5%</td>
<td>93.9%</td>
<td>4.6%</td>
</tr>
<tr>
<td>09:00 - 17:00</td>
<td>2</td>
<td>237</td>
<td>0.8%</td>
<td>95.2%</td>
<td>4.0%</td>
</tr>
<tr>
<td>10:00 - 18:00</td>
<td>2</td>
<td>128</td>
<td>4.6%</td>
<td>94.3%</td>
<td>1.1%</td>
</tr>
<tr>
<td>10:30 - 18:30</td>
<td>1</td>
<td>119</td>
<td>2.9%</td>
<td>97.1%</td>
<td>0.0%</td>
</tr>
<tr>
<td>11:00 - 19:00</td>
<td>1</td>
<td>37</td>
<td>6.7%</td>
<td>93.3%</td>
<td>0.0%</td>
</tr>
<tr>
<td>14:00 - 22:00</td>
<td>1</td>
<td>69</td>
<td>7.3%</td>
<td>92.7%</td>
<td>0.0%</td>
</tr>
<tr>
<td>15:00 - 22:00</td>
<td>1</td>
<td>14</td>
<td>9.1%</td>
<td>90.9%</td>
<td>0.0%</td>
</tr>
<tr>
<td>16:00 - 24:00</td>
<td>1</td>
<td>20</td>
<td>10.5%</td>
<td>89.5%</td>
<td>0.0%</td>
</tr>
<tr>
<td>22:00 - 06:00</td>
<td>1</td>
<td>9</td>
<td>0.0%</td>
<td>100.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

Optimized pool division We want to enforce a division of the vehicle fleet into pools. The first and second pool contain vehicles that transport only Covid–19 patients and no Covid–19 patients, respectively. A third pool is made up of floater vehicles. We evaluate the results and develop an approach how this pool division can be used in practice.

An example is given in Table 3. We collected all information about transports that took place in some area between 1st January, 2020 and 30th June, 2021. The percentage of Covid–19 transports is quite low, as days with very few (or even zero) transports are included. It stands out that floater vehicles are mostly used in the earlier shifts. Later in the day, it is more often possible to use pure Covid–19 vehicles. This might be caused by a higher number of transports in the morning, as discussed in Section 1. Moreover, we can gain an intuition which vehicles are a good choice for fixed Covid–19 vehicles using such tables. Depending on the number of Covid–19 transports on one day, some of these vehicles should be assigned to the Covid–19 vehicle pool.

Because the exact distribution of transports, i.e., how many infected patients should be transported, is not known in advance, it is helpful to decide on the number of vehicles for each pool depending on the current number of Covid–19 cases. We assumed previously that these numbers can have a high correlation. Furthermore, it can be helpful to hold an additional floater vehicle in reserve to allow for any discrepancies. Such an evaluation is again possible for different scenarios, i.e., areas and weekdays. Again, similar peculiarities will be obtained that lead to an improved pool distribution.

Comparison of the approaches We now compare the two possibilities discussed to handle Covid–19 transports. To do this, we select an example day in April 2020 when many infections occurred. In total, there are 45 requested transports, 17 of which are for Covid–19 patients. We solve the problem for both cases without using dummy nodes. Instead we treat semiplannable transports like ad–hoc transports. For the second approach, i.e., limiting the number of Covid–19 transports per vehicle, we assume that two sets of protective clothing are available for each vehicle. In total, there are little data containing a high percentage of Covid–19 transports. Thus, we consider an example that yields the same maximum and total delay for both heuristic approaches,
but the transports are handled very differently depending on the approach used. We will discuss these differences concerning the vehicle fleet.

In our example, 17 vehicles are available. In both schedules, 15 of them are used. Actually, both approaches use the same vehicles, only those with a shift time in the early morning are not needed as there is no requested transport.

If we minimize the number of Covid–19 vehicles, then three of the 15 vehicles used are full Covid–19 vehicles and five additional vehicles are used as floater vehicles. In the other case, there are four Covid–19 and six floater vehicles. The larger fleet is necessary (and not penalized) as the number of transports it can handle is limited. In fact, when minimizing the number of Covid–19 vehicles, some Covid–19 and floater vehicles handle at least three Covid–19 transports, which is hardly penalized when distributing the clothing equally.

One thing that stands out in both schedules is that every vehicle that transports any infected patient has all such transports at the end of its shift. So, the penalty $\gamma_{\text{change}}$ is never applied. The fact that this is possible with both approaches and also produces a relatively small delay is a good result in terms of minimizing infection risks. The largest delays occur for transports later in the evening and cannot be prevented even if we do not include any Covid–19 requirements besides the increasing transport duration. This comes due to the fact that there are not enough vehicles available at the same time.

In conclusion, both approaches have their merits and produce similar results most of the time, especially when the number of Covid–19 transports is quite low in practice. A decision for one approach can, for example, be made depending on the amount of protective clothing available. As soon as this number increases, the need to consider this limitation is reduced, enabling a focus on minimizing contacts between infected and non–infected patients and employees. Another idea is combining both approaches, i.e., to distribute the protective clothing to all vehicles but provide more to those that are likely to have more Covid–19 transports to do. For this, a fleet division like the one previously discussed could be helpful.

5 Conclusion

In this work, we have proposed a solution approach for scheduling non–urgent patient transports. Information about these transports is incomplete and may only be partly known several hours before they are required. Our objective is to minimize the delay for patients in a fair manner. We use a VRPGTW formulation that can then be solved by a state–of–the–art MIP solver.

We implemented the MIP formulation for the cases of full and incomplete information. We classify required transports into plannable transports (full information), semiplannable transports (full information but the target time is unknown) and ad–hoc transports (no information about the target time at all). Ad–hoc transports are handled by an iterative algorithm that solves the standard model every time that full information about a transport becomes known. Semiplannable transports can be covered by introducing dummy transports with an estimated target time and are then treated as plannable transports. It is also possible to treat them like ad–hoc transports. We have compared our modeling approach to the current scheduling practice of the dispatcher. Thereby, we observe that the waiting times in the optimized schedules are significantly lower than those obtained via a simulation of the current scheduling practice. To incorporate semiplannable transports where significant improvements can be seen, we require more data that includes such transports. Using the current data, we were only able to elaborate on some examples.

We have extended the model so that Covid–19 transports can be handled by different fleets. Still, the model remains solvable in real time and can be solved with MIP–based algorithms. We outlined algorithmic approaches, which speed up the solution process.

In summary, we proposed a formulation for the scheduling problem of patient transports that can be used in practice, also with further extensions not limited to the pandemic situation. With the availability of more data, it is expected that the proposed approach will work even better. Due to high security standards, currently the methods developed here cannot yet be used to schedule patient transports at the control centers.

Several research directions are of interest for the future. As already mentioned, the usage of multi–objective optimization might be helpful. Another potential for improvement lies in incorporating
semiplannable transports, where – assuming more data is available – other methods, e.g. further estimations of the duration, can be implemented. Furthermore, the usage of dummy nodes can be extended, so that they can be created for more types of transport than those presented here for dialysis. Our approach can also be transferred to different scheduling/routing problems, e.g., the taxi routing problem described in [14].

Acknowledgements

We are grateful for continuous support from the Integrated Control Center ILS, Nuremberg, and for many fruitful and stimulating discussions with Dr. Marc Gistrichovsky and Ludwig Fuchs (ILS). Research reported in this paper was supported by project HealthFaCT under BMBF grant 05M16WEC. Furthermore, this paper has received funding from the European Union’s Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 764759.

References


