Minimizing Delays of Patient Transports with Incomplete Information including Covid-19 Requirements

Dennis Adelhütte¹, Kristin Braun*¹, Frauke Liers¹, and Sebastian Tschuppik¹

¹Friedrich-Alexander-Universität Erlangen-Nürnberg, Department Mathematik, Cauerstraße 11, 91058 Erlangen

February 8, 2021

Abstract

We investigate a challenging task in ambulatory care, namely minimizing delays in patient transport. In practice, a limited number of vehicles are available for non-emergency transport. Furthermore, the planner rarely has access to complete information when establishing a transport plan for dispatching the vehicles. If additional transport is requested on demand, the schedules have to be updated, which can lead to long waiting times. We model this 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 different models that are designed to determine fair and stable plans that are protected against uncertainties. Furthermore, we show that the model can easily be modified when transports have additional requirements, such as an affiliation to different types that can for example have a different duration or need to be separated. This is demonstrated by incorporating requirements imposed due to the Covid-19 pandemic. To show the applicability of our approach, we conduct a numerical study using historical data and compare our optimization approach to existing scheduling approaches. This reveals that and show that, in general, mathematical optimization methods can significantly decrease the delay and have considerable potential for optimized transport schedules.

Keywords: OR in health services, Robust Optimization, Vehicle Routing, Online Optimization, Mixed-Integer Linear Optimization

1 Introduction

Healthcare systems comprise many components, for example emergency rescue transports or hospitals or doctor’s practices. While these are examples of crucial elements of healthcare, a significant number of vehicles are also required for non-emergency transport. These can be, for example, transports for patients who have been discharged from a hospital or patients receiving certain treatments. While in principle this transport could be carried out by the same vehicles that carry out rescue transports, in Germany a different fleet is typically used for non-urgent patient transport. In contrast to emergency rescue, this form of transportation does allow for delays, though this is clearly not desirable. When scheduling transport, a number of issues must be taken into account: The number of vehicles in the transport fleet is limited and the drivers’ work shifts necessarily 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.

In practice, the problem is usually solved with a greedy approach: Whenever a transport is requested, the vehicle that can reach the patient most quickly of all available vehicles is dispatched.

*Corresponding author
For the resulting transport plans, optimization potential is usually disregarded and sometimes, transports have even to be re-scheduled for the following day.

In this paper, we model the problem of finding fair schedules for patient transports: By 'fair', we mean mean that, as the first priority, the maximum delay in all patient transport is minimized. The aim is to distribute the minimal delay among the patients in a roughly equal fashion and whenever possible, the shift times are not violated. We present two approaches to handle the transport that is requested during the course of the day: If the planner has no knowledge of the requested transport in advance, but does not yet know the desired time (for example a patient needs to be taken home after treatment), a data-driven approach is used to estimate that time. Otherwise, we re-optimize the transportation that has not yet begun. The model is based on the Vehicle Routing Problem with General Time Windows that was introduced in [21]. We demonstrate that our optimized procedure is effective and show that the model can be modified whenever necessary: In the Covid-19-pandemic, additional issues have to be taken into account to minimize the risk of infection. In particular, it is desirable to transport patients with known infection using separate vehicles.

**Structure of the Paper**  
In Section 2, the patient transport problem is described on an abstract level. We first define different kinds of transportation and discuss the amount of information available at the time of planning. Then, we describe the planning and the operational phase. 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 different transportation, so-called semi-plannable and ad-hoc transports. We then describe how to incorporate Covid-19-related requirements, including disinfection time and the goal of separating known Covid-19 transport from other requests as far as possible. We then elaborate on algorithmic methods to improve the running time for our models’ solving process. In Section 4 and Section 5, we give algorithmic approaches and evaluate these using available historical data, respectively. For real historical data from a German control center, we provide some statistics about delays in the historical data before we show the efficiency of our proposed approach. In particular, we compare the method to an implementation simulating the current scheduling practice at a control center. It transpires that, not only does this method decrease the waiting times for patients, it is also able to reduce violations of shift time, which is relevant for legal reasons.

Taking Covid-19-requirements into consideration, we finally study the separation of vehicles into different fleets. Here, patients with known infection are transported within one fleet, while the second fleet takes the other patients. Some 'floater' vehicles may support both fleets. In Section 6, we conclude our results and provide some ideas for future research.

**Literature and methodology review**  
When facing uncertainty while planning future events, techniques from two different fields can be applied, namely stochastic optimization [9] and robust optimization [5]. For the first, an underlying probability distribution of the unknown data is usually required. In our case, this would be a probability distribution of the scheduled times or a probability distribution of transportation being requested at certain locations. The second one involves an attempt to optimize over a pre-defined uncertainty set that is designed to incorporate all scenarios against which protection is sought. Since a probability distribution is not given here, we apply a concept from the second field, namely Recoverable Robustness [23], in which timetables are planned and, if occurring scenarios render them infeasible, they are recovered. This is generally a successful concept for handling uncertainties in combinatorial settings [30]. A survey of robust combinatorial optimization under convex and discrete uncertainty can be found in [11]. The approach of \(k\)-adaptability, for determining \(k\) decision rules before the uncertainty realizes is studied in [10] and applied to a Capacitated Vehicle Routing Problem with uncertain costs [17]. Another approach in combinatorial optimization under uncertainty is 'restricting' the uncertainty set with respect to a pre-given budget. This approach was introduced in [7] and extended to non-linear problems under uncertainty in [2], involving a numerical study involving the Vehicle Routing Problem with General Time Windows. Finally, the Vehicle Routing with Time Windows under uncertain travel time has been studied in [3].
In online optimization, decisions are taken whenever new information, such as a new transport request, becomes available. For a survey about online algorithms we refer to [19]. While in robust optimization, we plan a reaction to uncertainties (or take decisions before the uncertainty is realized), in online optimization a new decision is taken whenever a new request comes in. We apply approaches from the two fields and also attempt to combine approaches to determine when a robust solution or an online solution should be selected. To the best of our knowledge, [8] is the first paper to establish a framework that attempts to unify the approaches of stochastic optimization, robust optimization and online optimization in the sense that the designer of models is flexible, depending on the nature of the data.

Important work in the field of patient transport in the European area is contained in [14], where the focus is on the situation in the Netherlands. Unlike in Germany, ambulances there 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. Consequently, we do not make use of this option. In [15], the authors describe the situation in Austria very well. However, the settings differ from the one presented here, since they describe the stationing of rescue vehicles or the periodic delivery of blood reserves.

Solution methods close to our work are [6] and [18]. The former also uses the solution of snapshots to handle ad-hoc transports and the latter introduces dummies to precautionarily schedule ad-hoc transports. We combine and extend both works here.

Concerning additional requirements due to the Covid-19 pandemic, [24] 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 [4], 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 [12] and solved using a genetic algorithm.

Finally, we refer to [27] for a general overview for Vehicle Routing Problems. There, an overview of different variations for solving Vehicle Routing Problems and its modifications are presented and evaluated, including branch-and-bound-algorithms, branch-and-cut-algorithms, set-covering-based algorithms and heuristics, among others. 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. For future research, it would certainly be interesting and beneficial to apply said algorithms to our models.

**Notation**

The set \( \mathbb{R} \) denotes the set of real numbers, \( \mathbb{R}_+ \) denotes the set of non-negative real numbers. With \( \mathbb{N} \), we denote the non-negative integer numbers and \( \mathbb{N}_{>0} := \mathbb{N} \setminus \{0\} \) is the set of positive integers.

2 Description of the Problem

In this section, we describe the patient transport problem in more detail. As already described, minimizing delays for patients waiting for their transport is our main focus. In the following, we first discuss feasible schedules.

2.1 Feasibility and Quality of Schedules

A transport of a patient involves the following data: A person has to be driven from one place to another, e.g. for treatment or after being discharged from a hospital and the transport is scheduled for a fixed time. A vehicle cannot transport more than one patient at the same time and the 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 $d_T$, which denotes the total time of $T$ and

iv) its target time $t_T$, which denotes the time when $T$ is supposed to be carried out.

A plannable patient transport is not considered an emergency, i.e. it does not necessarily have to be carried out at its target time. The set of plannable transports is denoted as $T_{\text{plan}}$. The planner has complete information about the set $T_{\text{plan}}$ 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 drivers of the vehicles have work shifts, a tour should start and end during the shift. This proves to be a bottleneck as we will see in our numerical study. Thus, we define the following:

Definition 2.2. Let $T_{\text{plan}}$ be a set of plannable patient transports and let, for all $T \in T_{\text{plan}}$, $y_T$ be a non-negative number, such that $y_T \geq t_T$. The set $S := \{(T, y_T) : T \in T_{\text{plan}}\}$ is called the schedule. The schedule is called feasible if

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

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

If only (i) holds, the schedule is called weakly feasible. The set of (weakly) feasible schedules is henceforth denoted as $X$ ($X_{\text{weak}}$).

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. However, the shift times of the vehicles 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. As other aspects, like costs of a trip or violating shift times as little as possible (or not at all) can be important as well, we define the following:

Definition 2.3. Let $X_{\text{weak}}$ be a set of weakly feasible schedules and assume we have $l \in \mathbb{N}$ criteria $g_1(x), \ldots, g_l(x)$, all defined on $X_{\text{weak}}$, to measure the quality of the schedules. Let $\gamma_1, \ldots, \gamma_l \geq 0$ be a set of weights. We call the function $f : X_{\text{weak}} \rightarrow \mathbb{R}$ 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 [16]. For future research, it would be interesting to solve the problem with a multi-objective approach.

Possible criteria include the maximum delay of all transports, 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

$$\min_{x \in X} f(x) \quad \text{or, if necessary,} \quad \min_{x \in X_{\text{weak}}} h(x). \quad (1)$$

The exact model is given in Section 3. We assume $h(x) := f(x) + g(x)$, with $g(x)$ measuring the violation of the working shifts. As in practice not all transports are known in advance, in the following we briefly describe the resultant uncertainties.

### 2.2 Planning with Incomplete Information

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 dialysis patient needs to be returned home after the dialysis but the dispatcher does not know when the treatment will be finished. Secondly, a transport is requested that was not known about when establishing a schedule for $T_{\text{plan}}$. For example, this could be patients that are discharged from the hospital after a doctor’s visit. 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 semi-plannable or uncertain 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 $d_T$, which denotes the total time of $T$.

For a semi-plannable transport, no target time is known. The set of semi-plannable transports is denoted as $T_{\text{semi}}$.

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

The notion of plannable, semi-plannable 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 [23]. In the planning phase, the transports of $T_{\text{plan}}$ are scheduled. After the planning phase, when the schedule is carried out, the operational phase begins. While the transports of $T_{\text{plan}}$ are carried out, transports of $T_{\text{adhoc}}$ become known and have to be incorporated into the schedule. Thus, the schedule usually has to be updated since a target time for ad-hoc-transports is requested.

The semi-plannable transports are a special case and can be planned in both phases: On the one hand, we can always treat them like unplannable transports and 'wait' until their target time becomes known, i.e., until they are requested. While this is straightforward, this approach has the down-side of ignoring information we already have, namely that the transport must take place. On the other hand, if we have enough data, we can treat them like plannable transports and estimate their target time. Further details are provided in Section 3.

This concludes the discussion of the patient transport problem and the solution approach. In the next section, we model the specific optimization problems and describe how to incorporate semi-plannable and unplannable transports into the model.

3 The Mathematical Model

The task from the previous section is now modeled as a Vehicle Routing Problem with General Time Windows (VRPGTW) that was introduced in [21]. Firstly, the VRPGTW is modeled as an MIP. Then, we explain that scheduling plannable transports only are modeled as a VRPGTW. Thirdly, semi-plannable transports are incorporated and the operational phase is described by including ad-hoc transports. Finally, we discuss treatment of Covid-19 transport.

3.1 The Vehicle Routing Problem with General Time Windows

The VRPGTW is defined on a directed graph $D := (V, A)$ with vertex set $V := \{0, \ldots, n\}$, $n \in \mathbb{N}$, arc set $A \subseteq V \times V$ and a set of homogeneous vehicles $K := \{1, \ldots, m\}$, $m \in \mathbb{N}_{>0}$. 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 the node 0 is the depot where all vehicles start and end their trip. The duration $\text{dur}_{i,j} \geq 0$ denotes the time that is required to serve customer $i$ and to drive to customer $j$ afterwards.

For all $(i, j) \in A, k \in K$, the binary variable $x_{i,j,k} \in \{0, 1\}$ indicates whether vehicle $k$ travels on the arc $(i, j)$ and for all $i \in N$, $y_i \in \mathbb{R}_+$ is the time when customer $i$ is served by vehicle $k$. 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 of customer $i$. The goal is now to serve each customer exactly once, with as little penalty as possible. The model can be described as a mixed-integer non-linear optimization problem (MINLP) that is easy to linearize, see Model (2). We minimize the penalties. Constraints (2b) – (2e) ensure that there is a tour where each vehicle starts and ends at the depot and each customer is served exactly once. Together with constraint (2f), valid tours are defined. Furthermore, this constraint ensures that a vehicle which serves customer $j$ directly after customer
VRPGTW is defined as follows:

Let us assume that the above parameters are sufficient. Furthermore, we model feasible routes for general vehicle routing problems, as performed in [27]. The VRPGTW vehicles also have a capacity for transporting goods, and other costs are involved as well. However, for our purposes, the above parameters are sufficient. Furthermore, we model feasible routes for general vehicle routing problems, as performed in [27].

### 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 homogeneous vehicles.
- $A_N := \{(i, j) \in N \times N : i \neq j\}$ 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 serving $i$.
- $A_K := \{(0, j) : j \in N\} \cup \{(i, n + 1) : i \in N\} \cup \{(0, n + 1)\}$ is the set of arcs from 0 to all $i \in N$ and 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 $D := (V, A)$ 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 $k \in K$, $[a_k, b_k] \subset \mathbb{R}_+$ denotes the shift of the drivers of vehicle $k$.
- For $i, j \in N$, $d_{i,j} \geq 0$ denotes the time a vehicle needs to reach $O_j$ after finishing transport $i$, i.e., the sum of the duration of transport $i$ and the travel time from $D_i$ to $O_j$.
- For $j \in N$, $d_{0,j} \geq 0$ denotes the travel time for vehicle $k \in K$ to reach $O_j$ from its depot and $d_{j,n+1} \geq 0$ denotes the travel time for vehicle $k \in K$ to reach its depot from $D_j$. We set $d_{0,n+1} := 0$ for all $k \in K$. 

$i$ finishes the service of customer $i$ and travels from customer $i$ to $j$ before the vehicle starts serving customer $j$.

$$\min \sum_{i \in N} p_i(y_i),$$

subject to

$$\sum_{k \in K} \sum_{j \in \delta^+(i)} x_{i,j,k} = 1 \quad \forall i \in N,$$

$$\sum_{j \in \delta^+(0)} x_{0,j,k} = 1 \quad \forall k \in K,$$

$$\sum_{j \in \delta^+(i)} x_{i,j,k} - \sum_{j \in \delta^-(i)} x_{i,j,k} = 0 \quad \forall k \in K, i \in N,$$

$$\sum_{i \in \delta^-(0)} x_{i,0,k} = 1 \quad \forall k \in K,$$

$$x_{i,j,k}(y_i + d_{i,j} - y_j) \leq 0 \quad \forall k \in K, (i, j) \in A,$$

$$x_{i,j,k} \in \{0, 1\} \quad \forall k \in K, (i, j) \in A,$$

$$y_i \geq 0 \quad \forall i \in N.$$
• $\gamma_{\text{max}} > 0$ is the weight for the maximum delay (in the objective).

The variables of our model are defined in the following:

• $x_{i,j,k} \in \{0, 1\}$ 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 resp. delays and the weighted maximum delay are the (piecewise linear) measures of quality, i.e.

$$\gamma_{\text{max}} \cdot \max\{0, y_1 - t_1, \ldots, y_n - t_n\} + \sum_{i \in N} \max\{0, y_i - t_i\}. \quad (3)$$

Thus, the (linearized) instance of the VRPGTW is Model (4). Here, we introduced the shift times as hard bounds that necessarily need to be satisfied in constraints (4e) and (4f) since we aim for feasible schedules. In constraint (4g), we further ensure that a vehicle does not start a transport before the transport is scheduled.

With Model (4), we can consequently establish a feasible schedule: If $x_{i,j,k} = 1$, 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 $(0, v_{k_1}, \ldots, v_{k_s}, n+1)$, 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:

$$\min \gamma_{\text{max}} \cdot \mathcal{D} + \sum_{i \in N} \delta_i, \quad (4a)$$

s.t. Constraints (2b) – (2e),

$$y_i + d_{i,j} - y_j \leq M_{i,j}(1 - x_{i,j,k}) \quad \forall k \in K, (i,j) \in A_N, \quad (4b)$$
$$y_{i,k} + d_{i,j} - y_{j,k} \leq M_{i,j}(1 - x_{i,j,k}) \quad \forall k \in K, (i,j) \in A_K, \quad (4c)$$
$$a_k \leq y_{0,k} \quad \forall k \in K, \quad (4d)$$
$$y_{n+1,k} \leq b_k \quad \forall k \in K, \quad (4e)$$
$$t_i \leq y_i \quad \forall k \in K, i \in N, \quad (4f)$$
$$y_i - t_i \leq \delta_i \quad \forall i \in N, \quad (4g)$$
$$0 \leq \delta_i \leq \mathcal{D} \quad \forall i \in N, \quad (4h)$$
$$x_{i,j,k} \in \{0, 1\} \quad \forall k \in K, (i,j) \in A, \quad (4i)$$
$$y_{i,k} \geq 0 \quad \forall i \in N, k \in K. \quad (4j)$$

### 3.3 Incorporating Semi-Plannable and Ad-Hoc-Transports

As discussed in Subsection 2.2, two kinds of uncertainties have to be incorporated: transports with unknown target times and ad-hoc-transports. We start with the first type.

**Semi-plannable transports** For transports $T \in \mathcal{T}_{\text{semi}}$, no time is yet defined. Treating them as ad-hoc-transports, the planner ‘ignores’ such a transport until its target time becomes known. Alternatively, its target time could be estimated, denoted as $t_T^{\text{est}}$ and Model (4) could be solved for plannable transports and transports with estimated target time. However, the true target time may differ from the estimate. If the target time is earlier than estimated, the planner will be informed and can treat the request as an ad-hoc transport. If the target time is later than estimated, the scheduled vehicle would have to wait. Since the target time is only estimated, the actual target time could be much later than the estimated time, 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 3.1. A transport $T$ is called a dummy transport, if

- $T \in T_{semis}$, i.e., it is semi-plannable,
- its target time is estimated, henceforth denoted as $t_{est}^T$, and
- it has a waiting time $w_T \geq 0$ after which the transport will be treated as an ad-hoc-transport.

The set of dummy transports is denoted as $T_{dummy}$.

To estimate the target time, historical data or medical expertise can be used.

Remark 3.2. The introduction of dummy transports has already been used in e.g. [18] for solving routing problems for the delivery of urgent goods that are subject to online input. Applying clustering methods and estimating target times, they applied the same idea that vehicles wait at certain places for a certain time.

Now, dummy transports are treated as plannable transports, i.e., the VRPGTW instance Model (4) is solved. 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 do not count (i.e., only the delay of the plannable transports are relevant). Secondly, the individual delay of dummy transports is a quality of measure and is weighted by $\frac{1}{2}$. This can be further evaluated but at the moment it is only important that the weight is smaller than 1. And finally, if its actual target time is higher than its waiting time, the quality of the schedule decreases. Thus, the objective of Model (4) is

$$\gamma_{max} \cdot D + \sum_{i \in N} \delta_i + \frac{1}{2} \sum_{i \in T_{dummy}} \max\{0, t_i - t_{est}^i\} + \sum_{i \in T_{dummy}} M_i \max\{0, t_i - w_i\}$$

with $M_i \in \mathbb{R}_+$ being a sufficiently large number. We note that this objective can easily be linearized.

Ad-hoc-transports

With the planning of all transports in $T_{plan}$, the planning phase is completed. The planner is called whenever an ad-hoc transport has to be scheduled at a certain time $\sigma$ on the fly. Since at this point of time $P^0$ is already carried out, some vehicles are already on their trips. Furthermore, serving this ad-hoc transport using a vehicle in the depot 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 $V^\sigma$ denotes all nodes that are represented in the model before the new ad-hoc transport is requested. The set $P^\sigma := \{(i, j, k, y^*_{i,k}) \in V^\sigma \times V^\sigma \times K \times \mathbb{R}_+ : x^*_{i,j,k} = 1\}$ denotes the schedule that is carried out at $\sigma$. If no ad-hoc-transports have been requested before $\sigma$, $P^\sigma = P^0$. Transports already in operation are kept unchanged. They are denoted by the set $P^\sigma_{fixed} := \{(i, j, k, y^*_{i,k}) \in P^\sigma : \sigma \geq y_{j,k} - d_{i,j}\}$ that contains all transports, where the vehicle is at least on the way from $D_i$ to $O_j$. Now, if an ad-hoc-transport is requested for time point $\sigma$, we do not change any transports in $P^\sigma_{fixed}$ but, instead, 'remove' every transport in $P^\sigma \setminus P^\sigma_{fixed}$. 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 (4) with the modification that all transports in $P^\sigma_{fixed}$ are unchanged. We update $D$ by including the new ad-hoc-transport in $V^\sigma$ and $A^\sigma$, respectively.

To ensure that fixed transports are not changed, we introduce two additional constraints, namely

$$x_{i,j,k} = x^*_{i,j,k} \forall (i, j, k, y^*_{i,k}) \in P^\sigma_{fixed} \hspace{1cm} (5)$$

and

$$y_{j,k} = y^*_{j,k} \forall (i, j, k, y^*_{i,k}) \in P^\sigma_{fixed} \hspace{1cm} (6)$$

The basic model can be solved with MIP-solvers. In the following, we describe how to modify the models if transports have further restrictions during the Covid-19 pandemic.
3.4 Handling of Covid-19 Transports

It is desirable that transportation is divided into two types, depending on whether a patient is (suspected to be) infected with Covid-19 or not. As true for any serious infectious illness, in Covid-19 transport vehicles the staff has to wear protective clothing and the vehicle is disinfected after every transport. This scenario has been investigated in e.g. [25]. The additional time this takes must be taken into account when establishing the schedules.

Since the provision of protective clothing is limited and in order to reduce unnecessary contact, we present two approaches, namely reducing the number of vehicles that are allowed to carry Covid-19 patients and limiting the number of Covid-19 patients per vehicle. In both cases, we still aim to minimize the waiting times for patients. These different objectives are handled by different penalty parameters in the objective function of Model (4) and the parameter values need to be chosen appropriately.

Minimizing the Number of Covid-19 Vehicles An option to reduce contact between uninfected and infected individuals is to divide the vehicle fleet into different pools, one pool of vehicles that only handle Covid-19 transport and a pool for all other forms of transport. It is possible to additionally use so-called floater vehicles that are able to do both types of transports to maintain a 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.

Limiting the Number of Covid-19 Transports for each Vehicle A different approach is to distribute the protective clothing equally between all vehicles. In this case, every vehicle is able to serve a limited number of Covid-19 transports and then it has to return to the depot to get 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 transport and so some of them may be handled with a smaller delay than when separating the fleets completely. Nevertheless, this modelling approach can result in an increased number of switches between infected and non-infected patients for each vehicle. Using historical data, we will evaluate both approaches in Subsubsection 5.2.4 and provide some statistics.

Mathematical Formulation The additional effort for Covid-19 transport can be modeled as a constant time \( t_{cor} > 0 \) that is added to \( d_{i,j} \), if transport \( i \) is known to involve Covid-19. The information on whether a transport must be handled as such is given in the input data as a binary parameter \( c_i \) for each transport \( i \in N \). This time is needed after each transport where a (potentially) infected patient was transported. Thus, to handle the Covid-19 situation, independently of the actual scheduling procedure, we replace constraint (4c) with

\[
x_{i,j,k} \cdot (y_{i,k} + c_i \cdot t_{cor} + d_{i,j} - y_{j,k}) \leq 0 \quad \forall k \in K, (i,j) \in A_N.
\]  

Its linearization is given by

\[
y_{i,k} + c_i \cdot t_{cor} + d_{i,j} - y_{j,k} \leq M_{i,j} (1 - x_{i,j,k}) \quad \forall k \in K, (i,j) \in A_N.
\]

Similarly for \( A_K \) only \( d_{i,j} \) changes to \( d_{i,k} \). For both proposed approaches, we further introduce binary variables \( u_{i,j,k} \in \{0,1\} \) for \( i,j \in N \) and \( k \in K \). These indicate whether some non-infected patient \( j \) is transported immediately after a (potentially) infected patient \( i \) using the vehicle \( k \). Thus, it is modeled by

\[
u_{i,j,k} \geq (c_i - c_j) \cdot x_{i,j,k}, \ k \in K, (i,j) \in A_N
\]

and penalized using \( \gamma_{change} \) in the objective function.

To minimize the number of Covid-19 vehicles, i.e., those that transport Covid-19 patients, we introduce binary variables \( \lambda_k \in \{0,1\} \) for all vehicles \( k \in K \). They indicate whether a vehicle is used as a Covid-19 vehicle at some point during the day. Therefore it is modeled by imposing the inequality

\[
\lambda_k \geq c_i \cdot x_{i,j,k}, \ k \in K, (i,j) \in A
\]

with a further penalty \( \gamma_{veh} \) for using a vehicle as a Covid-19 one.
As previously mentioned, the models presented in the earlier sections are solved via state-of-the-art MIP solvers that use branch-and-cut approaches, cf. [29], [28] or [22]. We also introduce the constraint

\[ \min \gamma_{\text{max}} \cdot D + \sum_{i \in N} \delta_i + \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} \gamma_{\text{change}} \cdot \mu_{i,j,k} + \sum_{k \in K} \gamma_{\text{change}} \cdot \lambda_k \]  

s.t. Constraints (4b), (4e) – (4j),

\[ y_{i,k} + c_i \cdot t_{\text{cor}} + d_{i,j} - y_{j,k} \leq M_{i,j}(1 - x_{i,j,k}) \quad \forall k \in K, (i, j) \in A, \]  
\[ \lambda_k \geq c_i \cdot x_{i,j,k} \quad \forall k \in K, (i, j) \in A, \]  
\[ \mu_{i,j,k} \geq (c_i - c_j) \cdot x_{i,j,k} \quad \forall k \in K, i, j \in N, \]  
\[ \lambda_k, \mu_{i,j,k} \in \{0, 1\} \quad \forall k \in K, (i, j) \in A. \]

This concludes the modeling of the patient transport problem. In the next section, we present the solution algorithm and present a numerical study to show the efficiency of our approaches.

4 Solution Algorithms

As previously mentioned, the models presented in the earlier sections are solved via state-of-the-art MIP solvers that use branch-and-bound approaches, cf. [29], [28] or [22]. We also develop some heuristics, some of which are integrated in the solver in order to achieve a better performance. We will explain the algorithms in the following.

4.1 Implementation of a Simulated Reality

To evaluate the optimized schedules, we require a baseline solution. Directly comparing the results obtained here to historical schedules from the past leads to some problems. Firstly, some of the

For the second approach, we introduce an integer variable \( u_k \) that indicates if and how often protective clothing needs to be restocked for vehicle \( k \) and, thus, it has to return to the depot. Therefore, we introduce the constraint

\[ (u_k - 1) \cdot \alpha \geq \sum_{(i,j) \in A} c_i \cdot x_{i,j,k} \]  

for all vehicles \( k \in K \). The parameter \( \alpha \) indicates how much protective clothing a vehicle initially has on board. It is possible to use different values per vehicle but in our case we assume that this number is fixed. It is also possible to determine these parameter values depending on solutions that are created using a pool division.

To implement the limitation of protective clothing, we use a penalty parameter \( \gamma_{\text{clothing}} \) that is multiplied by \( u_k \). This means the penalty is applied every time a vehicle has to return to the depot to get new clothing. The aim is that this is never necessary and thus the penalty is chosen to be quite high so a vehicle only exceeds its limit if it is unavoidable. The model would be more complex whether we implemented the return to the depot explicitly. For example, it would be necessary to check if there is new clothing available at the depot that might be returned by another vehicle. Instead, we model this approach in a way that does not necessarily require return drives during the shift. After including the new penalty parameter, the model is then given by

\[ \min \gamma_{\text{max}} \cdot D + \sum_{i \in N} \delta_i + \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} \gamma_{\text{change}} \cdot \mu_{i,j,k} + \sum_{k \in K} \gamma_{\text{clothing}} \cdot u_k \]  

s.t. Constraints (4b), (4e) – (4j),

\[ y_{i,k} + c_i \cdot t_{\text{cor}} + d_{i,j} - y_{j,k} \leq M_{i,j}(1 - x_{i,j,k}) \quad \forall k \in K, (i, j) \in A, \]  
\[ \mu_{i,j,k} \geq (c_i - c_j) \cdot x_{i,j,k} \quad \forall k \in K, i, j \in N, \]  
\[ (u_k - 1) \cdot \alpha \geq \sum_{(i,j) \in A} c_i \cdot x_{i,j,k} \quad \forall k \in K, \]  
\[ \mu_{i,j,k} \in \{0, 1\} \quad \forall k \in K, \]  
\[ v_k, u_k \in \mathbb{N} \quad \forall k \in K. \]
historical data is incorrect or missing. During preprocessing, this was handled by either estimating or more often by deleting the corresponding transports. A second point relates to practical decisions made by humans, e.g., renting vehicles between different catchment areas or using them for rescue transport in the worst case. Another human decision is to postpone some transport, possibly to a later day. Such decisions are not contained in the model described here.

We therefore implement a baseline simulation for the decisions made by the dispatcher, working on the same data, with the same restrictions. The obtained schedules are used as a baseline comparison to evaluate our solution. In practice, a greedy approach for urgent transport is currently used for all kinds of transport. 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 must be adhere to. 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 estimate whether the vehicle could reach its depot without violating its shift times.

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

4.2 Enhancement of the MIP Solver

Now, we present some algorithmic ideas to speed up the MIP solution process for the models from the earlier sections. From real historical data, we use a small subset of our data, namely 59 days (January and February) from two areas, i.e. 118 instances in total. The areas have different sizes and we evaluate our algorithms on them. The smaller area has about 100,000 inhabitants, the larger area about 500,000.

Using Primal Heuristics The first idea is to include primal heuristics that aim 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 solution of the LP relaxation. The number of rounded variables is chosen such that the effort for rounding does not exceed the resultant gain in speed. 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 MIP solution. Here, we take the number of vehicles $|K|$ and set the $|K|$ greatest variables to 1. If a feasible solution is found by using this start solution, it is used for the following solving process. Otherwise, it is discarded. It is important not too fix too many variables as the solution may then become infeasible. In contrast, fixing too few variables will not speed-up the solving process.

Using a Greedy Heuristic as First Feasible Solution The first feasible solution is derived from the simulation of the current scheduling approach given in Subsection 4.1. This greedy simulation can be computed very quickly. Every schedule that is obtained by this baseline is (at least weakly) feasible and we can construct a solution for the variables $x_{i,j,k}$ – the remaining values can easily be calculated by the solver. This solution is then used as a start solution. The solution is used once as a known feasible solution when the MIP solver is started.

Determining a Branching Priority Within a branching step, a variable is chosen from all binary and integer variables, depending on some measure that aims to create preferably small trees, see for example [1]. Let us consider the determination of optimized plans under Covid-19 restrictions. If the values for the binary variables $x_{i,j,k}$ are decided, the unique values for the remaining binary and integer variables are logically implied and can easily be derived. Thus, we obtain a branching priority and branch on the former variables first that we give as input to the
Table 1: Comparison of different speed up methods for smaller problems in the upper table and larger problems in the lower table. There are 59 problems each.

<table>
<thead>
<tr>
<th></th>
<th>Primal heuristic</th>
<th>Greedy heuristic</th>
<th>No heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Average time for solved problems</strong></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><strong>Average time for solved problems</strong></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>

solver. Possibilities for improving the solving process of the VRPGTW, e.g. determining bounds, are given in [13].

**Handling of Large Instances**  Especially on days where many transport requests are made, the solving process of the resulting MIPs can take a long time. 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 as a schedule at this time. To ensure that a (weakly) feasible solution is always found, we give a start solution, namely the solution that can be calculated using the greedy algorithm from Subsection 4.1. The idea behind this is that urgent decisions, i.e., those that must be made in the near future, can be dealt with directly while longer-term decisions can be rescheduled at a later stage.

In the following section about numerical results, we concentrate on limiting the total running time because it ensures that we obtain optimal solutions. However, we also evaluate the effectiveness of directly limiting the time for each MIP.

### 5 Implementation and Numerical Results

In this section, we provide details on the implementation, as well as statistics and results that can be obtained from the solution process, i.e., from instances as well as optimized schedules. Secondly, in Subsection 5.1, we provide general statistics about transportation in Middle Franconia. Finally, we evaluate how the performance of the approach when incorporating semi-plannable transports and considering the Covid-19 pandemic in Subsection 5.2.

The models and algorithms were implemented in Python 3.7.7. To solve them, we used Gurobi 9.0.2 [20] 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. We solve each day separately, and, unless otherwise mentioned, we specify a time limit of one hour as we figured out that a higher time limit of e.g. 24 hours does not make a significant difference. We will elaborate on this later in Subsection 5.2.

**Comparison of the heuristic methods**  In Table 1, both heuristic approaches are compared to each other and to the results if no such heuristic is used. We evaluate on both an easier set and a harder set with larger instances. In the smaller test set around 20 transports per day are requested, while in the larger test set there are up to 65 transports per day. During the weekend, the problems are smaller. The number of vehicles is between 5 and 20, depending on the weekday and the size of the area. The number of binary and continuous variables can be estimated by

\[ O(|K| \cdot |V|^2) \text{ and } O(|K| \cdot |V|), \]  

respectively. The number of constraints is given by

\[ O(|K| \cdot |V|^2). \]
5.1 General Statistics about the Requested Patient Transports

We provide some general statistics about the transport data. We consider the number of plannable and ad-hoc transports over the year and accumulated over one day. In Figure 1a, we plot the number of patient transports from January until August 2020. There are weekly minima corresponding to the weekends, especially to Sundays, while the overall number of transports has a similar magnitude. As the year goes on, there is a small drop starting in April. In previous years there is no such drop, so we can assume that this was caused by the beginning of the Covid-19 pandemic.

To better understand pandemic requirements on patient transports, we divided all transports on the basis of whether a transport features a (potentially) infected patient. 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. It can be seen that in the peak phase up to 30% of all transports were classified as (suspected) Covid-19 cases. The percentage of transports is quite similar to the number of Covid-19 cases in Germany, cf. [26].

Furthermore, we examine the number of transports during the course of a day in Figure 2. Here, transports are distinguished depending on their target time and the number of transports per hour is plotted from January to August 2020. As can be seen 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 once more relatively low.
In addition, we consider the number of plannable and ad-hoc transports separately. The number of plannable transports is given in the gray bars, while the black bars represent the number of ad-hoc transports. In general, 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 while in the morning most of the transports are plannable. However, this is not true in the early morning hours, i.e., between 6 and 8 am.

A possible reason for the increasing number of ad-hoc transports in the afternoon are relocations that are required after some examination in the hospital. Furthermore, this statistic regards return trips with previously unknown target time as ad-hoc transports. Their number also increases in the afternoon.

5.2 Improvements using the VRPGTW Model

In the following, we compare the historical data’s solution and the existing planner’s course of action with our optimization approach.

5.2.1 General Model with Plannable and Ad-Hoc Transports

Now, we evaluate the optimized schedules against the baseline solution of Subsection 4.1. Unless otherwise mentioned, semi-plannable transports are treated as ad-hoc transports. As an example, we show the results 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 the time limit.

In Figure 3, these are compared for two different regions. 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 where 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 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. On this day, many transports were requested for the late afternoon, showing the potential of our optimization approach.
Figure 3: Improvement in maximum delay that can be gained using our optimization, when compared to the simulated reality. If a difference is highlighted in red, this means that an improvement in shift time violations could be achieved.

We consider the improvement in the lower plot that considers 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. It can be seen that the optimized schedule often decreases the maximum delay by up to one hour. Furthermore, on two days, there is a reduced number of shift-time violations as well as a smaller maximum delay. (We recall that we respect shift times in our model and only violate them if otherwise no feasible schedule could be obtained.) The average improvement in total delay is around 10 minutes per day.

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 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 optimum for the transports already known, it could lead to a better complete plan if 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 a single problem is aborted and the current best solution 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 sub-areas, i.e., such with up to 130 transports.
per day, with more vehicles and transports. In the following, we evaluate the schedules that result. 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. 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. For the remaining problems, we improve the maximum delay 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 schedules are compared to the optimal solution. To determine this, we first evaluate how many problems could be solved to optimality. Figure 5 shows the number of instances where a minimum percentage of MIPs were solved completely. For example, in 289 of 364 instances, we found an optimal solution for at least 80% of all MIPs. Secondly, in Figure 6a, the quality of the solutions of aborted MIPs is given. To do this, we group the solutions by their gap between primal and dual solutions. This gap is computed directly by Gurobi and for minimization problems it is given by

\[ g = \frac{z_P - z_D}{z_P} \]  

where \( z_P \) and \( z_D \) are the current primal and dual objective bounds. In our case, a dual bound is always given by at least 0 and so we know that \( 0 \leq g \leq 1 \). Furthermore, in Figure 6b, we give the gap for the same instances but with a time-limit of 60 minutes.
In both cases, most problems create a gap that is near to the bounds, i.e., 0 or 1. For the time-limit of 10 minutes, there are few more problems that create a very small gap, it is very likely that they can be solved during 60 minutes. The same is true for a gap near 1, where it could at least be improved. If a gap near 1 occurs, this means that the dual bound could not really be improved, as, in our problems, we always have a trivial lower bound of 0. This also happens often in both cases. This suggests that these problems are very hard to solve and will still take a very long time until an optimal solution is found. Many of these problems are MIPs that are (presumably) only weakly feasible. Here the objective has two goals, namely minimizing the delay as well as the shift time violations, which increase the difficulty of the problem. For the practical usage, we, nevertheless, recommend to use this approach because it still improves the current course of action and can be computed in a reasonable time.

5.2.2 Insights from the Optimal Solution

In Subsubsection 5.2.1, we evaluated the optimization results and discovered that they improve the current course of action. Now, we wish to draw further conclusions from these schedules that can help to create better prerequisites for example, be a recommendation for how shift times should be changed such that an adjusted vehicle fleet can prevent delays in advance.
First, we examine how the delays vary over the course of the day. In Figure 7a, the maximum and average delay aggregated over the year is given for every hour. The average delay is calculated using the number of transports. In Figure 7b, we further plotted 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 as we will discuss later. The average delay is very high in the early morning and decreases continuously. To gain a better impression 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, there are also very 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 start not earlier than 6am and they first have to drive to the start point 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 are probably not able to handle these transport requests without risking shift time violations.

We state that changing some shift times to enable the period between 5am and 7am to be handled better can decrease waiting times in this period. Nevertheless, we cannot decide what is precisely feasible because there are also employment laws and logistical aspects that have to be considered.

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 then have a knock-on effect on 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.

5.2.3 Incorporating Semi-Plannable Transports

We evaluate the usage of dummy transports for semi-plannable transports. In practice, there are some problems with the given data so that the desired results cannot be seen very often. First, transports are divided depending on the county of the origin location. Dialysis centers are often located in big cities but many patients live in rural areas. These patients therefore have to be transported between different areas and transports are not represented in the same model. This leads to dummies that are never matched by the actual return journey. A second problem is that some data is missing or incorrect. 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 journey 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, i.e., it is a plannable transport, the return journey also looks plannable in our data, even if it is not.

Now, we consider a schedule where dummy transports actually decreased the total and maximum waiting times. Table 2a and Table 2b display two schedules that were created on a Wednesday. 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 the 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, especially the actual schedule time.

Table 2b shows the schedule created at 11:00. As can be seen, the vehicles allocated to the non-fixed transports 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 larger improvement can be expected if more dummies could be used, i.e., if appropriate data is available.
Table 2: 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>y</th>
<th>y_i,k</th>
<th>y_i,k + s_i</th>
<th>δ_i</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>3.8629</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>10:30</td>
<td>10</td>
<td>11:23</td>
<td>12:22</td>
<td>43.1832</td>
</tr>
<tr>
<td>11</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>y</th>
<th>y_i,k</th>
<th>y_i,k + s_i</th>
<th>δ_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>06:45</td>
<td>0</td>
<td>06:45</td>
<td>07:54</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>3.8629</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>10:30</td>
<td>10</td>
<td>11:23</td>
<td>12:22</td>
<td>43.1832</td>
</tr>
<tr>
<td>11</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>

5.2.4 Handling of Covid-19 Transportation

Until now, the dispatcher has no 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 could be handled using a similar approach to that in Subsubsection 5.2.2. This is accomplished for the first approach assigning the vehicles into pools. In particular, we evaluate this pool division and how it can be used in practice.

Optimized Pool Division With Model (11), we enforced 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 so-called floater vehicles that are

<table>
<thead>
<tr>
<th>Shift time</th>
<th>Number of vehicles</th>
<th>Number of Covid-19 transports</th>
<th>Covid-19 vehicle</th>
<th>Non-Covid vehicle</th>
<th>Floater vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td>06:00 - 14:00</td>
<td>1</td>
<td>97</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>07:00 - 15:00</td>
<td>1</td>
<td>98</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>08:00 - 16:00</td>
<td>1</td>
<td>79</td>
<td>6.7%</td>
<td>90%</td>
<td>3.3%</td>
</tr>
<tr>
<td>08:30 - 16:30</td>
<td>1</td>
<td>62</td>
<td>0%</td>
<td>96.5%</td>
<td>3.5%</td>
</tr>
<tr>
<td>09:00 - 17:00</td>
<td>2</td>
<td>85</td>
<td>1.9%</td>
<td>96.2%</td>
<td>1.9%</td>
</tr>
<tr>
<td>10:00 - 18:00</td>
<td>2</td>
<td>61</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>10:30 - 18:30</td>
<td>1</td>
<td>34</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>11:00 - 19:00</td>
<td>1</td>
<td>39</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>14:00 - 22:00</td>
<td>1</td>
<td>19</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>15:00 - 22:00</td>
<td>1</td>
<td>15</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>16:00 - 24:00</td>
<td>1</td>
<td>11</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>22:00 - 06:00</td>
<td>1</td>
<td>6</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Figure 8: Optimized pools for vehicles in one area and the example shift times for Tuesdays. Such tables can be generated for all weekdays and shift plans.
allowed to handle both types of transport. Now, we want to evaluate the results and develop an approach how this pool division can be used in practice. An example is given in Figure 8. We collected all information about transports that took place in some area between January and August 2020. The percentage of Covid-19 transports is quite low, as days with very few (or even zero) transports are included. Nevertheless, we can see that the Covid-19 and floater vehicle pools always use vehicles with similar shift times, namely between 8am and 5pm. We can say that these vehicles are good candidates for a fixed Covid-19 pool. 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 the number of vehicles for each pool depending on the current number of Covid-19 cases. We mentioned before that these numbers closely correspond. 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 wish to compare the two possibilities mentioned to handle Covid-19 transports. To do this, we select an example day in April when many infections occur. In total, there are 45 requested transports, 17 of which are for Covid-19 patients. We solve this problem for both cases without using dummy nodes. Instead we treat semi-plannable transports in the same way as 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 is little data containing a high percentage of Covid-19 transports so we will not recommend one approach over the other. Instead, we consider an example that yields the same maximum and total delay for both approaches, but the transports are handled very differently depending on the approach used. We will discuss these differences concerning the vehicle fleet.

On the example Saturday, 17 vehicles were available. In both schedules, 15 of them are used. Actually, both approaches use the same vehicles, only those with a shift time from 22:00 to 6:00, starting the previous day, are not necessary because there is no transport requested earlier than 6:00.

If we minimize the number of Covid-19 vehicles, three of the 15 vehicles used are full Covid-19 vehicles, while five additional vehicles are used as floaters. In the other case, there are four Covid-19 and six floater vehicles. This larger fleet is necessary (and not penalized) as the number of transports it can handle is limited. In fact, for the first approach, some Covid-19 and floater vehicles handle at least three Covid-19 transports, which is hardly penalized in the opposite approach.

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 for \( \mu_{i,j,k} \) is never applied. The fact that this is possible with both approaches, and further generates a delay that remains reasonably small, is a good result concerning the minimization of infection risks. The largest delays happen 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.

In conclusion, we can say that both approaches have their merits and – most of the time – produce similar results, especially if the number of Covid-19 transports is quite low. 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. A further idea is to combine 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 can be helpful. Thus, one can adjust Model (13) such that \( \alpha \) depends on the vehicle \( k \).
6 Conclusion

In this work, we proposed a solution approach to scheduling 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 VRPGRTW formulation that can then be solved by an MIP.

We implemented the MIP formulation for the case with full, as well as the case with incomplete, information. We classify required transports into plannable transports (full information), semi-plannable transports (full information but the target time is unknown) and ad-hoc transports (no information 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. Semi-plannable 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, i.e., ignore the given partial information until full information becomes known. These cases have been evaluated and compared to the current scheduling practice. 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 semi-plannable transports where significant improvements can be seen, we need more data that includes such transports. Using the current data, we were only able to elaborate on some examples.

We extended the model so that Covid-19 transports can be handled by different fleets as far as possible. Still, the model remains algorithmically tractable and can be solved via MIP methods. 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 high security standards, currently the methods developed here cannot yet be used to schedule patient transports at the control centers. It would be highly desirable to be able to use the tool in practice.

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 semi-plannable transports, where – assuming sufficient 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 [18].

Acknowledgements

We are grateful for continuous support from the 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 Sklodowska-Curie grant agreement No 764759.

References


