OR Spectrum
https://doi.org/10.1007/s00291-025-00812-3

ORIGINAL ARTICLE

The dial-a-ride problem in primary care with flexible scheduling

Felix Rauh[1,2] · Emma Ahrens[1] · Christina Büsing[1] · Martin Comis[1] · Felix Engelhardt[1]

Received: 31 August 2023 / Accepted: 5 February 2025

Abstract

Patient transportation systems are instrumental in lowering access barriers in primary care by taking patients to their general practitioners (GPs). However, the economic sustainability of such transportation systems based on ride sharing strongly depends on how well transportation requests can be bundled. We consider a dial-a-ride setting where the transportation requests consist of a ride to the GP and back. Patients may be chronic or "walk-in" patients, with the latter requiring transportation on short notice. In the general setting, the GPs fix appointments without consideration of the transportation. In our flexible scheduling setting, for chronic patients only an appointment range is fixed a priori, and the exact time is determined when the vehicle routes are computed. To tackle this setting, we propose a novel extension of the dial-a-ride problem, the dial-a-ride problem with combined requests and flexible scheduling (DARPCF). We introduce a heuristic for the DARPCF, called MCLIH, that is designed to exploit this increased flexibility. Initially, MCLIH computes so-called mini-clusters of outbound requests. Then, the mini-clusters are linked by solving a traveling salesman problem and creating routes of outbound rides with a splitting procedure. Our computational study shows that in rural regions with MCLIH and the flexible scheduling of chronic appointments, the average number of served transportation requests can be increased by 38% compared to a non-flexible setting.

Keywords Dial-a-ride problem (DARP) · Heuristics · Patient transportation · Primary care

1 Introduction

The aging population in rural areas is facing increasing barriers accessing primary care services (Syed et al. 2013). On the one hand, public transportation systems are often poorly developed and impractical for visiting a general practitioner

Extended author information available on the last page of the article

Published online: 09 April 2025

(GP) (Berg and Ihlström 2019). On the other hand, individual mobility decreases with age and the use of a cab is generally expensive (Ahern and Hine 2012).

To offer an efficient and convenient alternative, we explore the use of so-called *dial-a-ride systems* for transporting patients to GPs. These systems provide the comfort of direct transportation between patients' homes and GPs, while achieving greater cost-efficiency than cab services by pooling transportation requests. The standard process of arranging transportation in such systems is the following. First, a patient contacts their GP to arrange a fixed-time appointment, e.g., Monday 10 a.m. This time is then communicated to the transportation provider which in turn makes a commitment to take the patient from their home to the appointment and back. While this procedure is relatively comfortable for GPs and patients, it comes with the downside of not synchronizing appointment scheduling and patient transportation. As a result, geographically similar transportation requests may be widely spread in time, which prevents an efficient pooling and ultimately implies high transportation cost.

To alleviate this drawback, we propose a new concept for dial-a-ride systems that enables a partial synchronization of appointment scheduling and patient transportation. We thereby focus on appointments that are known several weeks in advance and introduce the so-called *flexible scheduling*. The idea of flexible scheduling is to change the arrangement of appointments and transportation: when a patient arranges an appointment with their GP, only a range for the appointment is fixed, e.g., Monday morning. The transportation provider can then flexibly schedule and reschedule the patient's rides within the previously agreed range. Finally, the transportation provider fixes the vehicle routes a few days ahead of transportation and thereby determines the exact appointments which are then communicated to patients and GPs.

While flexible scheduling clearly offers the potential to reduce transportation cost, it requires a certain degree of flexibility from both patients and GPs. For patients, this means that the exact appointment times are confirmed closer to the actual appointment date. This approach is particularly suited to non-urgent cases, such as individuals with predictable care needs and regular schedules, like weekly appointments. Similar to current medical practice, we define these patients as *chronic patients*. By contrast, we refer to the remaining (non-chronic) patients as *walk-in patients*, i.e., those with fixed-time appointments scheduled on shorter notice. These two patient types form the basis of the model and the terminology used throughout the paper. While flexible scheduling demands some spontaneity from chronic patients, it offers a key benefit: guaranteed transportation to and from appointments, addressing a challenge that might otherwise leave them uncertain about access to transport services.

From the GPs' perspective, we require the reservation of flexible appointment slots that will only be filled a few days in advance. As a result, physicians always know if there remain gaps in their schedules, however can only start filling them with fixed-time appointments or walk-in patients shortly beforehand. Such provider prescribed restrictions on how available slots may be filled are a common concept in appointment scheduling (Gupta and Denton 2008).

Ideally, the GP and the transportation provider are integrated into a dedicated organizational structure, such as a Primary Care Network in the UK,[1] that coordinates healthcare services, aligns financial incentives for insurers and others, and streamlines patient transportation in the spectrum between non-specialized private cabs and high-priority emergency transport.

As part of our proposed concept, we allow walk-in patients to request transportation for the specific times of their appointments on short-notice. However, the transportation provider may turn these requests down if they do not fit into the current vehicle routes. Moreover, each request comprises two rides: an *outbound* ride from the patient to their GP and an *inbound* ride from the GP back to the patient's home. Once a transportation request has been accepted, it must be serviced entirely, i.e., neither the outbound nor the inbound journey may be canceled.

In this paper, we investigate the extension of the classical dial-a-ride problem (DARP) with combined outbound and inbound rides per request by flexible scheduling. To that end, we introduce the *dial-a-ride problem with combined requests and flexible scheduling* (DARPCF) that allows a flexible scheduling of the outbound rides. Each flexibly scheduled outbound ride entails a subsequent inbound ride with a fixed time window. The resulting problem is both $\mathcal{NP}$-hard and hard to solve in practice, as it is an extension of the classic DARP problem (Parragh et al. 2008).

As our main contribution, we present a heuristic called *Mini-Cluster Linking and Insertion Heuristic* (MCLIH) for the DARPCF that enables transportation companies to use flexible scheduling. MCLIH consists of two phases: First, the requests of chronic patients in the DARPCF setting are processed and second, the requests of walk-in patients are inserted by an online algorithm. The procedure starts by clustering the chronic patient requests. MCLIH then solves a traveling salesman problem (TSP) and creates routes of outbound rides with a splitting procedure and greedily inserts inbound rides. Finally, we compare the performance of MCLIH with exact and other heuristic methods on realistic test instances. Because of the large number of requests in the considered scenarios, the solution quality is not evaluated by the length of the vehicle routes but rather by the number of served customers. We show computationally that flexible scheduling can accommodate between 16−38% more patient requests compared to traditional systems, depending on the topology of the instances considered.

The remainder of this paper is structured as follows: We start by reviewing relevant literature in Sect. 2. Then, in Sect. 3, we formally introduce the DARPCF, followed by a description of the solution approach in Sect. 4. The results of our computational study are presented in Sect. 5. Finally, Sect. 6 concludes with a short summary and outlook on future work.

[1] See https://www.england.nhs.uk/primary-care/primary-care-networks/.

2 Literature review

We begin by giving a general overview of dial-a-ride problems and algorithms, followed by a discussion of existing work on integrating scheduling and routing in healthcare.

2.1 State-of-the-art

Dial-a-ride problems belong to the most classical optimization problems in transportation. For a detailed review on models and techniques, we refer to the extensive surveys of Molenbruch et al. (2017) and Ho et al. (2018). Additionally, Cordeau et al. (2023, 2024) provide two more specialized and recent surveys focused on attended home delivery and service problems, a DARP subclass with applications in home healthcare.

Problem formulation The DARP is a special case of the general pickup and delivery problem, where transportation must be facilitated between different customers with paired demands (Parragh et al. 2008). A key property is the introduction of different user-oriented service constraints such as maximum ride times and time windows. The seminal work by Cordeau and Laporte (2003) has become the standard setting for the DARP (Molenbruch et al. 2017). We present this problem definition in Sect. 3. Moreover, Cordeau (2006) proposed a widely-used MILP formulation, which serves as the foundation for the mixed-integer linear programs we use, see Appendices C, D.

Exact methods The first exact approach for the single-vehicle DARP was published in 1980 by Psaraftis (1980) who solved the problem with up to nine users through dynamic programming. However, the algorithm does not consider time windows but only so-called *maximum position shifts*. The first Branch-and-Cut algorithm using different types of valid inequalities was proposed by Cordeau (2006).

According to the extensive survey by Molenbruch et al. (2017), the most efficient exact algorithm known at that time (having solved all of Cordeau's instances with up to 96 users) was a Branch-and-Cut-and-Price method by Gschwind and Irnich (2015).

Heuristics and metaheuristics The first work on the DARP by Wilson et al. (1971) was further improved by Jaw et al. (1986) and by Madsen et al. (1995). The latter proposed the so-called REBUS algorithm, a fast insertion heuristic which can also process requests interactively, thus solving the *dynamic* DARP with time windows on departure or arrival. REBUS is a greedy algorithm that builds the schedule successively by inserting transportation requests in a fashion such that the cost function is minimized. An essential concept used is that of the *time slack* of a stop in the schedule, which is the largest increase of the departure time that is possible without violating any time window constraints. We use a greedy insertion heuristic (GIH) based on REBUS both as a baseline and as part of our solution approach, see Sect. 4.

Bodin and Sexton (1986) first proposed cluster-first route-second methods in 1986. After using a clustering method for assigning requests to vehicles, a heuristic single-vehicle solution is computed, which involves a Benders decomposition and a so-called *space-time heuristic*.

The notion of mini-clusters which we use in this work was introduced by Desrosiers et al. (1988) and describes a geographically cohesive set of requests that is served in one vehicle route segment. The authors propose an algorithm where mini-clusters are generated by using neighboring criteria and the routing problem is solved by column generation. Ioachim et al. (1995) improved this approach by applying a column generation algorithm in the mini-clustering phase.

An adaptive large neighborhood search metaheuristic that is claimed to be competitive with all state-of-the-art heuristics has been published by Gschwind and Drexl (2019). Recently, Gaul et al. (2021) solved a DARP problem with 500 requests using a rolling time-horizon approach and employing both ejection chains (Glover 1996; Curtois et al. 2018) and the ruin-recreate principle (Christiaens and Vanden Berghe 2020).

2.2 Integration of scheduling and routing in healthcare

A related field in healthcare where scheduling and routing are combined is home healthcare (HHC), as presented in a review by Fikar and Hirsch (2017). The approaches in that field also aim at combining different decision levels in order to improve operations. In the regular setting, this comprises organizing shifts, the assignments of nurses to patients, and routing decisions. However, the resulting routing problem does not compute a route with pick-up and delivery of patients as in the DARP, but rather minimizes the operator's routing cost or other objective functions without any kind of ride sharing or relevant vehicle capacity constraints (e.g., Cappanera and Scutellà (2015); Grenouilleau et al. (2019)). Compared to our work, another main difference is that, in HHC, capacity of routes tends to be limited by the combined time needed for transportation and serving patient requests. On the contrary, in our setting, capacity is the physical vehicle capacity and service at the patient location only consists of the time needed for pickup/delivery.

More closely related to vehicle routing but without joint scheduling, a recent paper by Adelhütte et al. (2021) considers patient transportation with incomplete information and semi-plannable transports, i.e., the outbound trip to the treatment is given with complete information, whereas the return trip has initially unknown time windows. The authors formulate a vehicle routing problem with general time windows and show in their numerical study that incorporating semi-plannable transport reduces waiting times compared to the previous scheduling method.

A similar result can be found in a paper by Schilde et al. (2011) who consider a DARP setting where stochastic information on whether an outbound request causes a corresponding inbound request is taken into account. The used algorithms are modifications of the two metaheuristic approaches of variable neighborhood search (VNS) and multiple plan approach (MPA) (e.g., Mladenović and Hansen (1997); Bent and Van Hentenryck (2004)). In the first, the stochastic information about the

return trips is used for comparing candidate solutions. The MPA extension uses sample return trips for extending initial solutions and eventually removes the return trips again in order to produce gaps for the insertion of the actual return trips.

Finally, Johnn et al. (2021) integrate multiple uncertainties concerning home service assignment, routing, and appointment scheduling. While their HHC problem setting differs from the one in this work, they use a similar idea of incrementally specifying the specific time of appointments.

2.3 Summary and contributions of the paper

In contrast to previous literature, we propose a novel setting in which the the decision maker can control both routing of patients to GPs and the respective appointment scheduling. Since patients. rather than personnel, are transported in this setting, new challenges are introduced: vehicle capacity and ride-sharing become crucial considerations–differing from most home healthcare contexts. Furthermore, the GP appointments impose a special structure, where fixing one initial outbound ride introduces constraints on the inbound ride. This also differs from the work on attended home delivery, as noted above.

We not only introduce this novel problem setting, but also provide an algorithm suitable to solve realistic instances including walk-in demands in practice. For that, we use two established heuristic principles from literature: REBUS and mini-clusters. We show how those can be combined to solve outbound sub-problems to optimality, and then add inbound routes via an insertion heuristic based on REBUS. Our computational experiments quantify the benefits of adopting the proposed setting and applying our algorithm.

In the next section, we describe the aforementioned problem setting and formulation, formally introduce the set of *chronic patients*, and define the more flexible approach to handling their appointments and associated transportation requests.

3 Problem description and notation

We begin this section by introducing the standard problem setting for the static (i.e., offline) dial-a-ride problem as presented by Cordeau and Laporte (2003), which was adopted by most authors in recent years Molenbruch et al. (2017). Then, we discuss the DARPCF and its alterations to the DARP through the concept of flexible scheduling. For an overview of all notation used in this work, we refer to Appendix A.

3.1 The static dial-a-ride problem

Let $n \in \mathbb{N}$ denote the number of requests (or patients) to be served. We define the DARP on a complete directed graph $G = (N, A)$, called the *DARP road graph*, where $N = P \cup D \cup \{0, 2n+1\}$, $P = \{1, 2, \dots, n\}$ and $D = \{n+1, n+2, \dots, 2n\}$. By $R = \{r_1, \dots, r_n\}$, we denote the set of all requests where each request

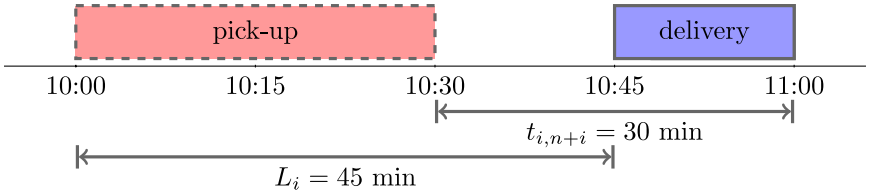


Fig. 1 Implicit pick-up time window for outbound ride $r_i = (i, n+i) \in R$ with direct travel time $t_{i,n+i} \in \mathbb{R}_{\geq 0}$ and maximum user ride time $L_i \in \mathbb{R}_+$

$r_i = (i, n+i) \in R \subseteq P \times D$ is represented by a pick-up node $i \in P$ and a drop-off (delivery) node $n+i \in D$. The nodes 0 and $2n+1$ represent an origin and a destination depot for the fleet of $m \in \mathbb{N}$ homogeneous vehicles. We denote by $t_{ij} \in \mathbb{R}_{\geq 0}$ the non-negative travel time between nodes $i \in N$ and $j \in N$. We associate a load $q_i \in \mathbb{Z}$ with each node $i \in N$, by default $+1$ for pick-up nodes and -1 for delivery nodes with a non-negative service duration $d_i, i \in N$, where $d_0 = d_{2n+1} = 0$. The time window for a node $i \in N$ is denoted by $[e_i, l_i]$ for $e_i, l_i \in \mathbb{R}, e_i \leq l_i$ and has a maximum length of $W \in \mathbb{R}_{\geq 0}$, i.e., $l_i - e_i \leq W$.

Moreover, let $Q \in \mathbb{N}$ denote the passenger capacity of each vehicle and $T_V \in \mathbb{R}_+$ the maximum route length for each vehicle. The maximum user ride time $L_i \in \mathbb{R}_+$ of request r_i for $i = 1, \ldots, n$ describes the maximum time the user may be on the vehicle for the ride. It can either be constant for all $i \in \{1, \ldots, n\}$ or proportional to the direct ride time, e.g., $L_i = 1.5 \cdot t_{i,n+i}$.

As done by Cordeau and Laporte (2003), we assume that for outbound trips only a time window for the delivery is specified, while for inbound trips there is only a time window for the pick-up. All other time windows are not specified explicitly, but derived implicitly from the maximum and minimum ride time constraints. For example, consider a customer requesting an outbound ride with a direct ride time of 30 minutes and a maximum ride time of 45 minutes. If the time window for the delivery is set to [10:45, 11:00], the implicit time window for the pick-up would be [10:00, 10:30] because no other pick-up time could result in a feasible ride; compare Fig. 1.

An optimal solution consists of a set of m vehicle routes from node 0 to node $2n+1$ such that for each request r_i for $i = 1, \ldots, n$ the nodes i and $n+i$ are contained in the same vehicle route in the correct order (the so-called *precedence constraint*). Moreover, the capacity and service constraints need to be satisfied and the routing cost must be minimized.

3.2 DARP with combined requests and flexible time windows (DARPCF)

We extend the above problem setting to the flexible scheduling of chronic patients. User requests now consist of two rides: an *outbound trip* to the appointment without time window, i.e., a time window which comprises the whole service period, and an *inbound trip* which has to take place within a time window of length W that starts when the stay at the GP of constant duration $d_{\text{GP}} \in \mathbb{R}_{\geq 0}$ ends.

The adaptation of the discussed model is straightforward: The set $R_c \subseteq P \times D$ only includes the chronic outbound requests. For the corresponding inbound trips, we introduce new vertex sets $\overline{D}$ and $\overline{P}$ which are copies of D and P, respectively. The inbound trip belonging to outbound trip $r_i = (i, n+i)$ is now given by $\overline{r_i} = (\overline{n+i}, \bar{i}) \in \overline{D} \times \overline{P}$. Thus, the set $\overline{R_c} := \{\overline{r_1}, \ldots, \overline{r_n}\} \in \overline{D} \times \overline{P}$ denotes the set of chronic inbound requests. The travel times for nodes j, k and their copies $\bar{j}, \bar{k}$ are set canonically to

$$t_{j\bar{j}} := 0, \quad t_{\overline{jk}} := t_{\bar{j}k} := t_{j\bar{k}} := t_{jk}.$$

The load of a copy is the negative of the original vertex, i.e., $q_{\bar{j}} = -q_j, j \in P \cup D$, and the time window for request $\overline{r_i}$ is denoted by $[e_{\bar{i}}, l_{\bar{i}}]$.

Definition 1 (DARPCF) Let the set of requests R consist of n pairwise outbound and inbound requests, i.e., $R = R_c \cup \overline{R_c} \subseteq P \times D \cup \overline{D} \times \overline{P}$ and let the DARP road graph $G = (N, A)$ with $N = P \cup D \cup \overline{D} \cup \overline{P} \cup \{0, 2n+1\}$ be the complete directed graph which contains a pick-up and a delivery node for each request. There is a homogeneous fleet of m vehicles with start depot at node 0 and end depot at node $2n+1$. Furthermore, d_{GP} denotes the duration of stay at the GP and W is the time window length.

We define the *DARP with Combined Requests and Flexible Time Windows (DARPCF)* as the problem of finding m vehicle routes on G which serve the pairwise requests in R in the following fashion: If an outbound request r_i is scheduled to arrive at time t_i, then the departure time $\overline{t_i}$ of the corresponding inbound request $\overline{r_i}$ must satisfy $\overline{t_i} \in [t_i + d_{\mathrm{GP}}, t_i + d_{\mathrm{GP}} + W]$. Moreover, the vehicle capacities, maximum route length and the maximum user ride time must be respected.

This extension makes the problem more complex since time windows of corresponding outbound and inbound trips are linked: We do not allow long waiting times between the outbound and inbound ride of a patient, or even worse, to schedule the inbound trip before the outbound trip. Moreover, the two rides may be scheduled on different vehicles.

In the definition above, we only model the requests of chronic patients. In a next step, we extend this definition to walk-in patients. In that context, we introduce the overall objective of the problem.

3.3 Extended DARPCF with walk-in requests

In the reality of primary care, a considerable amount of patients is not scheduled a long time in advance. In the following, we refer to these patients who have short-notice (non-flexible) appointments as *walk-in patients*. Their transportation requests are also pairwise outbound and inbound requests, and the corresponding sets are denoted by R_w and $\overline{R_w}$, respectively.

Given the appointment time s_i of the patient with requests $(r_i^w, \overline{r_i^w}) \in R_w \times \overline{R_w}$, we assume that the time windows for delivery at the GP and departure at the GP are given by $[e_i, l_i] = [s_i - W, s_i]$ and $[e_{\bar{i}}, l_{\bar{i}}] = [s_i + d_{\mathrm{GP}}, s_i + d_{\mathrm{GP}} + W]$, i.e., the outbound

delivery needs to be at most W before the appointment, and the inbound pick-up at most W after the end of the appointment.

Definition 2 (Extended DARPCF) Let $R = R_c \cup \overline{R_c} \cup R_w \cup \overline{R_w}$ be a set of requests consisting of pairwise requests of both chronic and walk-in patients, and let the DARP road graph and the other parameters be given as in the DARPCF. We define the *Extended DARPCF* as the problem aiming to find vehicle routes which maximize the number of served requests in the following fashion:

(i) Chronic requests $R_c \cup \overline{R_c}$ are scheduled flexibly as in the DARPCF.
(ii) Walk-in patients are served by scheduling the out- and inbound requests $(r_i^w, \overline{r_i^w}) \in R_w \times \overline{R_w}$ within the given time windows and by satisfying all other service and routing constraints as in the DARP.

In the reality of a healthcare service with limited budget, it is likely that the fleet is not sufficient to serve also all chronic and walk-in requests. Therefore, we explicitly allow to reject chronic requests as well. Nonetheless, it can be assumed that they are implicitly prioritized over walk-in patients due to their flexibility. Thus, the minimization of the number of rejected patients becomes the objective rather than total distance minimization.

Moreover, this problem definition assumes that the problem is solved only once when all requests are known. In fact, our algorithms schedule the two request types (i) and (ii) of the Extended DARPCF in separate stages and allow for further extensions that we briefly introduce in the following.

Further extensions We benchmark our approaches with an integer program for the Extended DARPCF which is laid out in Appendix D. However, our approaches also satisfy stricter constraints and can be applied in other problem variants which we summarize in the following.

- *Online requests:* Each walk-in request $(r_i^w, \overline{r_i^w}) \in R_w \times \overline{R_w}$ has a release time τ_i (potentially during the service period), and the operator must either accept and schedule or reject the request. When accepting the request, the resulting schedule $\mathcal{S}$ must remain the same for all stops before τ_i.
- *Non-continuous opening hours (lunch breaks):* Arrivals at the GP may not be scheduled during a specific time window during the days and must therefore be assigned either to the morning or the afternoon session.
- *Congestions at GPs:* Given parameters $C_{GP}, W_{GP} \in \mathbb{N}$, the schedules must satisfy that, when dividing the service period into intervals of length W_{GP}, not more than C_{GP} patients arrive within each of those intervals.

4 Solution methodology

In the following, we introduce a solution procedure for the Extended DARPCF, namely the *Mini-Cluster Linking and Insertion Heuristic* (MCLIH). For another procedure that we developed, a rolling horizon approach with specialized

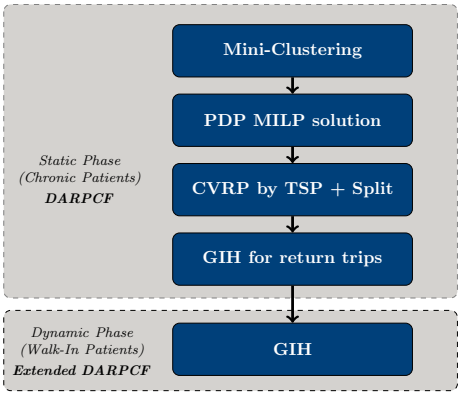


Fig. 2 The solution procedure MCLIH

matching problems called *Mini-Cluster Matching Algorithm*, we refer the reader to Büsing et al. (2021). MCLIH consists of two phases: In the first phase, all chronic patient requests are scheduled and in the second phase, all walk-in requests are added.

To obtain a schedule for the chronic patient requests, MCLIH uses a greedy mini-clustering algorithm which we present in Sect. 4.1. The algorithm calculates a partition $\mathcal{V} = \{\mathcal{M}_1, \ldots, \mathcal{M}_{|\mathcal{V}|}\}$ of the set R_c of all chronic outbound trips. Subsequently, an optimal route within each of these so-called *mini-clusters* is computed. Finally, the mini-clusters need to be linked to obtain the daily route for every vehicle.

As a linking procedure, MCLIH solves a capacitated vehicle routing problem (CVRP) on the mini-clusters $\mathcal{V}$. To that end, it links the mini-clusters to a TSP tour that is then split into the respective vehicle routes. Subsequently, a *greedy insertion heuristic* (GIH) based on the REBUS algorithm (Jaw et al. 1986; A heuristic algorithm 1995) is used to insert the chronic inbound trips. This GIH explores all possible insertions into the existing schedules and greedily chooses the one that minimizes a prespecified metric.

Eventually, MCLIH solves the online extension of the DARPCF by using the GIH for the insertion of walk-in patients. An overview of the different subroutines in MCLIH can be found in Fig. 2. Moreover, we include a pseudocode for MCLIH in Appendix B.

4.1 Mini-clustering

We start by introducing the notion of a *mini-cluster* which is a set of requests meant to be served by a single vehicle such that the vehicle is empty before and after serving the requests and such that no other request is served by that vehicle in this period. In our case, the mini-clusters initially only consist of up to $|Q|$ outbound requests, thus allowing us to relax the capacity constraint for the intra-cluster routes. We define a measure of how close two outbound requests $r_i, r_j \in R_c$ are by considering a directed graph $G' = (V, E)$: we create one vertex $v \in V$ for each outbound request $r \in R_c$ and introduce arc costs $c'_{ij} \in \mathbb{R}_{\geq 0}$ according to the

travel time of the shortest possible way to serve the requests r_i and r_j by starting at the pick-up location i of request r_i. Only the edges between requests which are profitable to combine are added, i.e., for some proximity parameter $\rho > 0$, it must hold that $c'_{ij} \leq \rho \cdot (t_{i,n+i} + t_{j,n+j})$. As the departure points differ, c'_{ij} and c'_{ji} are usually not equal and therefore the edge costs are asymmetric. The mini-clusters on G' are then computed by using a simple greedy algorithm similar to Kruskal's algorithm which grows mini-clusters (trees on G') until their size reaches the vehicle capacity Q.

Note that the definition of c' is useful specifically due to the flexible scheduling concept. In the classical DARP setting, two requests that lie close to each other are frequently not compatible due to their time windows. Here, outbound trips can be scheduled freely within a time slot and thus this problem disappears.

Optimal routes within mini-Clusters by solving MILP

Next, we compute for each mini-cluster an optimal route satisfying all patient requests within the considered cluster.

In comparison to the standard DARP formulation, we can omit the capacity and maximum route length constraints since the capacity constraint is satisfied due to the size of the mini-clusters and the maximum route length is checked afterwards when linking the mini-clusters (see Sect. 4.2). Moreover, no time windows need to be respected since the mini-clusters only consist of flexible outbound trips. The resulting problem is the classical Pickup and Delivery Problem (PDP, also referred to as pickup-delivery TSP (Kalantari et al. 1985)) with an additional maximum user ride time constraint. As the problem sizes are small, we can solve the PDP exactly by using an extension of the open TSP formulation as discussed, e.g., in Parragh et al. (2008).

4.2 TSP and split approach with GIH for return trips

Let us now consider the next step in the MCLIH procedure, which generates routes consisting of the mini-clusters computed in the preceding section.

We consider the graph $H = (\mathcal{V}, \mathcal{A})$ where each mini-cluster $\mathcal{M} \in \mathcal{V}$ corresponds to a vertex and where the arc costs $c_{ij} \in \mathbb{R}_{\geq 0}$ for $\mathcal{M}_i, \mathcal{M}_j \in \mathcal{V}$ are set to the travel time from the last drop-off location in the optimal route $\mathcal{S}_i$ on $\mathcal{M}_i$ to the first pick-up location in $\mathcal{S}_j$. We interpret the problem as a capacitated vehicle routing problem (CVRP) with slightly adjusted constraints compared to the standard formulation (Labadie 2016).

The basic idea of finding vehicle routes $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_m$ starting from a depot is the same as in the CVRP, however, we observe that our problem is set on a directed graph with asymmetric arc costs. Moreover, there is no actual capacity constraint to be considered because the vehicles are empty between mini-clusters. The route duration $d_{\mathcal{M}}$ within each mini-cluster $\mathcal{M}$ can be interpreted as service duration at that node. Thus, the important restriction for each vehicle, the maximum route duration, is of the form

$$\sum_{e \in A(\mathcal{R}_k)} c_e + \sum_{i \in V(\mathcal{R}_k)} d_i \leq T_V \tag{1}$$

for each route $\mathcal{R}_k$ for $k = 1, \ldots, m$. We can solve the problem in question by adapting CVRP algorithms to consider directed graphs and to respect equation (1) instead of the usual route length and capacity constraints.

Considering that it is a common and simple heuristic approach which has proven to produce competitive solutions, in particular for large instances (Prins et al. 2014), we decide to use the classical split method by Beasley (1983) in a more compact version proposed by Prins (2004) and Labadie et al. (2016). The method starts by solving a TSP on the vertices of H. To that end, we use an ant swarm TSP heuristic following the approach proposed by Dorigo et al. (1996).

The next step, splitting the TSP tour into sub-tours can be solved optimally in polynomial time by using a shortest-path algorithm on an auxiliary graph (Beasley 1983). The only necessary adaptations of the two algorithms to our setting are that the TSP needs to be solved with asymmetric edge costs and that the splitting procedure must respect equation (1). This can be done by modifying the CVRP capacity constraint by interpreting the service duration of a mini-cluster $\mathcal{M}$ as the load and by including the arc costs in the load calculation.

From the obtained vehicle routes we can deduce the times when the appointments of the chronic patients can start and thus determine the time windows for the inbound trips. By using the greedy insertion heuristic GIH based on Jaw et al. (1986) and Madsen et al. (1995) for the inbound trips, we finalize the schedules of the chronic patients. We assume that the number of vehicles is always large enough, such that this insertion is possible. However, it is also an option to remove an outbound request from the schedules in case of an unsuccessful insertion of the corresponding inbound trip and try reinsertion for different time windows.

5 Computational results

In this section, we describe the results of the study which we performed with our algorithm MCLIH. The aim is to verify whether or not the idea of flexible scheduling can improve a dial-a-ride system in primary care. Therefore, the focus lies on comparing the results in a flexible scheduling setting with the results in the standard setting. Moreover, we justify our choice to target rural areas by comparing realistic primary care settings in different regions. In particular, we describe in Sect. 5.1 how we generate instances for primary care in rural/semi-urban and urban areas, and lay out the obtained results in Sect. 5.2. Apart from that, we also illustrate the need for heuristic approaches and their performance by a comparison with a commercial solver.

Table 1 Considered regions for different degrees of urbanization

Region	Year	Population (≥16)	Type	Size [km^2]	GPs
Roetgen, Simmerath, Monschau	2011	35542 (29975)	rural	245.06	13
City of Pirmasens	2022	40403 (35584)	semi-urban	61.37	14
Monheim am Rhein	2022	43063 (37880)	urban	23.05	15

The number of adults is mapped to a 100 m square grid using aggregated regional data and the respective Community Identification Numbers

5.1 Data generation and study design

We evaluate the presented flexible scheduling approach based on real-world primary care systems in Germany. For that we consider three settings: a rural, a semi-urban and an urban one. Table 1 gives an overview of the regions chosen:

For that, we accessed data of the German National Census. The rural region is based on the 2011 census (Census 2011), as it is based on the existing model of the rural primary care system considered in the hybrid agent-based simulation tool SiM-Care (Comis et al. 2021). GP numbers and locations are the actual locations of GPs in the regions, based on publicly available data. As empirical transportation requests are unavailable, we resort to SiM-Care to generate artificial requests that reflect the system's structural characteristics. SiM-Care models both patients and GPs as individual agents and tracks their micro-interactions. As a result, we obtain the scheduled appointments of chronic and acute patients as well as the visits of walk-in patients for a one-year horizon. We set up two additional instances to model the effect of different topologies on the algorithm performance. For that, we used the current 2022 census data for Germany (Census 2022), and mapped the data to the region via the approach described in Comis et al. (2021). In line with Comis et al. (2021), we assume that children under the age of 16 will not make use of individual doctors appointments, these were excluded from the data.

Combining these with the locations of GPs, we generate the required outbound and inbound transportation requests that serve as the input to our models. We group these requests into instances that correspond to one day of simulated GP services for a sample of 13/14/15 GPs with a service period of six hours each. The number of pairwise requests per instance varies between 344 and 505. Table 2 shows the sample of 20 instances of the rural area corresponding to an arbitrarily chosen period of four weeks used in the study. The distances between the locations are based on road network data and computed by using the Open Source Routing Machine (OSRM) (Luxen and Vetter (2011).

Study design As described in the previous paragraph, we consider 20 instances, each comprising more than 360 patient requests. Since each patient request is due to a required consultation with a GP, it eventually results in one outbound trip and one inbound trip which have to be scheduled. In the flexible setting, represented by MCLIH, we discard the original appointment times of chronic patients that are provided from the simulation and re-schedule them throughout the whole service period.

Table 2 Patient structure in the considered instances of the rural setting and the algorithm performances on these instances with default parameter choices

	Instance (Number of requests)				Results (served requests)		
Day	Total	Chronic	Walk-in	Proportion chronic	GIHTW	GIHflex	MCLIH
490	349	310	39	0.89	165	116	254
491	472	254	218	0.54	170	144	225
492	475	259	216	0.55	182	162	229
493	487	256	231	0.53	176	164	235
494	457	255	202	0.56	166	161	241
497	373	313	60	0.84	160	117	227
498	469	276	193	0.59	175	149	222
499	502	233	269	0.46	175	159	231
500	466	233	233	0.50	171	149	233
501	453	242	211	0.53	164	146	233
504	354	307	47	0.87	164	122	236
505	445	266	179	0.60	165	138	232
506	491	261	230	0.53	169	151	232
507	479	243	236	0.51	159	158	217
508	432	240	192	0.56	165	147	214
511	344	318	26	0.92	142	133	237
512	456	263	193	0.58	168	141	227
513	484	248	236	0.51	162	162	212
514	469	248	221	0.53	165	168	236
515	438	235	203	0.54	164	161	229

As a reference for the standard setting we use GIH, the greedy insertion heuristic inspired by Madsen's REBUS algorithm (Jaw et al. 1986; Madsen et al. 1995). For GIH in the standard setting, denoted by GIHTW, we keep the original appointment times and create time windows before and after the appointment for the outbound trip and inbound trip, respectively. Recall that GIH is also part of MCLIH for the insertion of walk-in patients, therefore the main difference between the two settings is how many of the walk-in patients can still be inserted into the schedules after the static phase. In order to see if GIH can also profit from the flexible setting, we include it in a variant denoted by GIHflex. This means that the outbound trip can be inserted throughout the entire service period, and the subsequent insertion of the inbound trip needs to satisfy a time window derived accordingly.

In our study, we use the default values for the parameters depicted in Table 3. Moreover, we perform sensitivity analyses where single parameters like the maximum user ride time and vehicle capacity are varied to better understand the behavior of the algorithms.

Table 3 Default parameter choices

Parameter	Value	Information
m	10	Fleet size
Q	4	Vehicle capacity
W	20 min.	Maximum time window length
d_{GP}	30 min.	Duration of each stay at the GP
L_i	$1.5 \cdot t_{i,n+i}$	The maximum user ride time (proportional to direct travel time)
ρ	1.5	Measure of proximity of requests
$(C, W)_{GP}$	(6, 30 min.)	GP congestion parameters (each 30 min. max. 6 patients may arrive)

For the comparison with a commercial solver, we reduce the instance sizes by sampling subsets of requests from the complete instances described above. We use the Gurobi Optimizer in version 11.0.1 and perform experiments with a time limit of 60 min.

Implementation All algorithms presented in this work were implemented in Java (SE 17) and the experiments were performed on a linux computing environment (64 Bit) with 8 available Intel(R) Xeon(R) Gold 6240 CPUs clocked at 2.60GHz and 2 GB memory per CPU. The resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government.

We use the following external libraries or source codes: The routes between the different locations are calculated by using OSRM (Luxen and Vetter 2011). For the mini-clustering step, we modified the disjoint-set data structure and the Kruskal algorithm implementation by Esmer (2017). The MILP solution for the intra-cluster routes as well as the benchmark solutions are computed by using the Java API of the Gurobi Optimizer (version 11.0.1) (Gurobi 2023). The TSP tour that links the mini-clusters is computed by an ant colony optimization algorithm implemented by Dodd (2013) based on Dorigo et al. (1996).

5.2 Results

In this section, we present the results of our computational study. First, we evaluate which algorithm produces the best solutions in the default setting. Then, we analyze the algorithms' sensitivity to changes in the default parameter choices.

Main result

Based on the results depicted in Fig. 3 and included in Table 2, we can state that the introduced algorithm MCLIH employing the flexible scheduling of chronic patients outperforms GIH.

On average, across the 20 instances considered, MCLIH serves about 230 requests (mean: 230.1, median: 231), whereas GIHTW serves in excess of 60 requests fewer (mean: 166.35, median: 165). In fact, by using MCLIH and the flexible scheduling

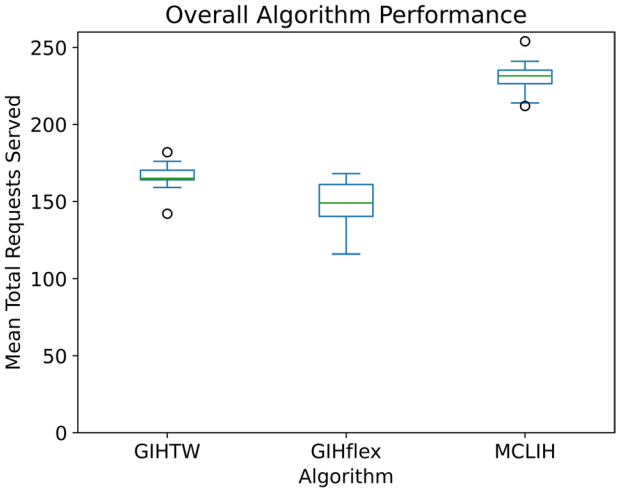
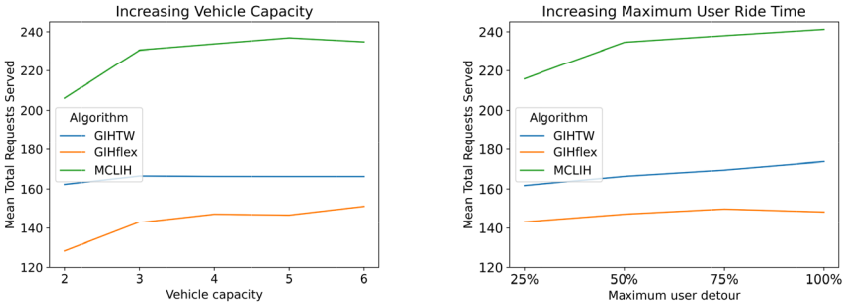


Fig. 3 Box plot for the number of served requests in the default setting



(a) Mean number of total requests served for different vehicle capacities.

(b) Mean number of total requests served for different maximum user ride times.

Fig. 4 Sensitivity of algorithm performances with respect to vehicle capacity and maximum user ride time

system, about 38% more requests can be served compared to the default system using GIHTW. Moreover, MCLIH gives good solutions consistently, with a standard deviation of 9.5.

The second observation is that using MCLIH in the flexible setting indeed makes a difference compared to using an algorithm like GIHflex which can schedule flexibly, but is not tailored to it. In fact, GIHflex cannot even outperform its non-flexible counterpart GIHTW, as it does not manage to group similar requests together efficiently and does not spread them over the entire service period.

Let us now perform some sensitivity analyses, and then look into further metrics and details of the setting.

Table 4 Comparison of route performance metrics across different topologies

Metric	Algorithm Setting	GIHTW	GIHflex	MCLIH
Total Requests Served	rural	1.0	0.89	1.38
(relative to GIHTW)	semi-urban	1.0	0.76	1.16
	urban	1.0	0.78	1.19
Total Route Duration	rural	1.0	0.72	0.97
(relative to GIHTW)	semi-urban	1.0	0.65	0.93
	urban	1.0	0.63	0.96
Average User Detour	rural	9.1%	12.7%	11.7%
	semi-urban	12.0%	15.0%	14.0%
	urban	10.0%	14.0%	13.0%

Sensitivity to vehicle capacity and maximum user ride time

We compare the performance of our algorithms for different vehicle capacities with values $Q \in \{2, 3, 4, 5, 6\}$ and different maximum user ride times L_i. The results are visualized in Figs. 4a and 4b.

Note that we limit the vehicle size to 6 as this keeps the run-time for the exact MILP solution within the mini-clusters mentioned in Section 4.1 in the range of seconds. In some cases for $Q = 6$, we use the current best intra-cluster route in case the MILP is not solved optimally after 30 s. The results are shown in Fig. 4a and we can conclude that an increased vehicle capacity does not generally imply that more requests are served.

We actually note that most requests are served by MCLIH for $Q = 5$, where the mean over all instances is 236.9. Analysis of the resulting schedules shows that even though we increase the vehicle capacity to 5 or higher, it hardly happens that 5 or more patients are on board of a vehicle at the same time. This suggests that the imposed capacity limit of $Q = 6$ does not pose a restriction in reality. One reason could be that the limit on the user ride time is a more limiting constraint during the schedule creation than the vehicle capacity, which we want to study in the following.

Therefore, we investigate how changes in the maximum user ride time L_i influence the resulting schedules. Reasonable choices for L_i include all values between 1.25 (i.e., 25% increase compared to direct ride) and 2 (rides can take twice as long). Figure 4b shows the results for different values of L_i while keeping all other parameters at their default setting.

We can observe a similar behavior as for increased Q, i.e., the three algorithms can profit from the increasing flexibility only to a small extent. For MCLIH, the mean number of served requests increases by about 11.7% when increasing the maximum user ride time from 1.25 to 2. This small increase does not reinforce our presumption that the number of served requests is more sensitive to the vehicle capacity than to the maximum user ride time. It seems more likely that the detour potential to combine requests is not larger because of the resulting inherent difficulty in providing the return trips within time. We now study how the chosen rural setting favors our approach.

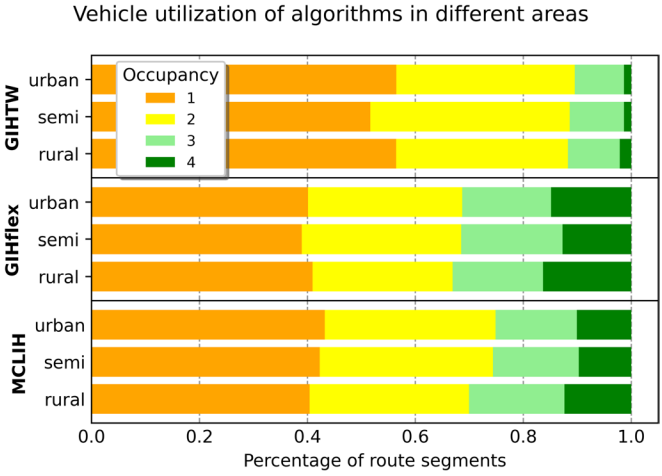


Fig. 5 Distribution of different vehicle occupancy levels across different topologies

Non-rural settings and further metrics Performing the experiments on the semi-urban and urban instances described in Table 1 demonstrates that the approach is particularly tailored to a rural setting. While the increase in performance in rural settings when utilizing MCLIH and flexible scheduling is 38%, Table 4 shows that is only 16 and 19% in the semi-urban and urban regions, respectively. Notably, MCLIH achieves these additional requests without extending the total route duration; in fact, there is a 2.8% decrease in the rural setting. However, this improvement is accompanied by a slight increase in the average user detour (i.e., the additional time patients spend on board compared to a direct ride, which is capped at a 50% in the default parameter setting). Specifically, patients experience an average detour of 9.1% under the GIHTW model, while this rises to 11.7% with the flexible setting of MCLIH. In non-rural topologies, these differences become even smaller.

For further understanding of MCLIH's performance, Fig. 5 provides an analysis of the maximum occupancy of each route segment between non-empty rides. In urban contexts with $Q = 4$, MCLIH reaches occupancy levels of 3 or 4 passengers in around 25% of it's route segments (i.e., the mini-clusters between empty rides). This value increase to approximately 30% in rural settings. In contrast, GIHTW reaches only 14.6% and 11.7% of segments with 3 or 4 passengers in these areas, respectively. A detailed examination of the schedules and metrics of GIHflex reveals that it initially accepts requests and creates efficient schedules with good utilization, but quickly struggles to insert further requests. The issue arises because requests are scheduled very tightly together in only a small part of the session as this minimizes the objective of the greedy insertion. However, this complicates the accommodation of the respective return trips as there is little space for insertion. This leads to the rejection of many requests and ultimately results in shorter total route durations.

Comparison with commercial solver When using Gurobi (with default settings and time limit of 3600 seconds), as soon as instances have 20 requests or more,

Table 5 Mean number of served requests for different instance sizes

n	8	12	16	20	24
Algorithm					
Gurobi	7.80	11.65	14.35	15.75	16.25
MCLIH	7.60	10.30	12.05	12.85	14.50
Ratio Gurobi/MCLIH	1.03	1.13	1.19	1.23	1.12

Compared to the default setting, we reduce fleet size and vehicle capacity to $m = 2$ and $Q = 3$, respectively. Moreover, the service period is reduced to three hours. When increasing the number of requests, two requests are added to the instances of the previous size. The only modification to default parameters of Gurobi involved setting MIP_HEURISTIC = 0.3 in order to find more feasible solutions

the problem cannot be solved to (proven) optimality in any of the cases, however, feasible solutions are always found. The results are depicted in Table 5.

We can see that, for $n = 20$, MCLIH serves on average 2.9 requests less which corresponds to the largest gap of 23%. When adding further requests, there are instances where either Gurobi or MCLIH can accommodate additional requests, however, no clear pattern can be identified. Considering that MCLIH's runtime in these situations is less than a second and thus three orders of magnitude smaller, the trade-off of solution quality to runtime of MCLIH appears promising. Note, however, that the solution properties on such small scale instances are different as also the vehicle fleet size and time horizon are reduced. Moreover, as mentioned in Sect. 3.2, the formulation (see Appendix D) does not satisfy certain additional constraints that we impose for MCLIH. Therefore, the true optimality gap of MCLIH may still be smaller. Overall, the size limitations of the exact method justify the usage of our fast heuristic approaches.

Summary The algorithm MCLIH shows that the number of transported patients can be considerably increased in a flexible scheduling setting. Our sensitivity analyses reveal that the overall behavior of MCLIH appears to be stable with respect to evaluated parameters. In fact, we observe that increasing the vehicle size can slightly improve the performance of MCLIH, but "overly optimistic" mini-clusters which make the assignment of the return trips inefficient may also occur. Increasing the maximum user ride time L_i (and similarly also the maximum time window W) generally helps improving the schedules to a small extent, but must be chosen moderately in a real setting. Using a commercial solver like Gurobi cannot be considered a realistic alternative due to its runtime. Moreover, we see that on small instances MCLIH performs almost as good as Gurobi.

6 Conclusion

In this paper, we introduced the DARPCF as a new extension to the standard DARP which has applications in customer transportation for customers with time flexibility. It allows flexible scheduling of the outbound ride and requires that the corresponding inbound ride is scheduled within a predetermined time after the first ride.

We developed an algorithm, MCLIH, which first creates mini-clusters of similar outbound rides and then links them to vehicle schedules in a route-first cluster-second method for capacitated vehicle routing problems. The method solves a TSP and splits the tour into sub-tours.

As the bases of our computational study, we generated transportation requests for a real-world rural primary care system using SiM-Care (Comis et al. 2021) and compared the new algorithms to the regular DARP setting when using a greedy insertion heuristic. Within our experiments, an increase of 38% of the number of served requests could be obtained in the rural setting, with notable but less pronounced improvements of 16−19% in the urban and semi-urban settings.

A limitation of the proposed flexible dial-a-ride system is its reliance on the flexibility of both customers and GPs. Consequently, the reported 38% increase in ride efficiency may not be sufficient to persuade operators to adopt new approaches to handling requests and appointments. In particular, networks of GPs with the authority and capacity to schedule appointments and coordinate patient transportation must be established and tailored to the regulatory framework of the respective healthcare system. Moreover, we see room for several improvements of presented algorithms, e.g., through more advanced mini-clustering techniques like a neighborhood search. Next to these improvements, future work should investigate algorithms for the DARPCF that further exploit the information that each outbound ride is followed by an inbound ride. Finally, we would like to investigate how historical data or probabilities for short-notice requests or appointment durations can be used to further improve the vehicle schedules.

7 Supplementary information

Not applicable

Appendix A: Notation

Table 6 summarizes the node sets in the DARP road graph, and Table 7 gives an overview of all further notation used, as defined in Sect. 3.1 and the following sections.

Recall that by choice of the indices, it holds that for a pick-up node $i \in P$, the node $i + n \in D$ corresponds to the delivery, and $\bar{i} \in \overline{P}$ to the respective inbound pick-up node in case of the Extended DARPCF. Furthermore, for the Mixed Integer Programs given below, we use $M_{ij}^k \geq \max\{0, l_i + d_i + t_{ij} - e_j\}$ and $W_{ij}^k \geq \min\{Q_k, Q_k + q_i\}$ for Big-M type constraints.

Table 6 Overview of the different node sets in the road graph $G = (N, A = N \times N)$ for the Static DARP and the Extended DARPCF. Note that each GP is represented by multiple vertices in the same geographical location, i.e., there is one copy per patient arrival and departure

Static DARP				
(Mixed out- and inbound)				
$N = P \cup D \cup \{0, 2n+1\}$				
Pick-up	Delivery			
$P = \{1, \ldots, n\}$	$D = \{n+1, \ldots, 2n\}$			
Extended DARPCF				
$N = P \cup D \cup \overline{P} \cup \overline{D} \cup \{0, 2n+1\}$				
	Outbound		Inbound	
	Pick-Up	Delivery	Pick-Up	Delivery
Chronic	P^c	D^c	$\overline{P}^c$	$\overline{D}^c$
Walk-In	P^w	D^w	$\overline{P}^w$	$\overline{D}^w$
Together	$P = \{1, \ldots, n\}$	$D = \{n+1, \ldots, 2n\}$	$\overline{P} = \{\overline{1}, \ldots, \overline{n}\}$	$\overline{D} = \{\overline{n+1}, \ldots, \overline{2n}\}$

Table 7 Overview of notation used in this work

Parameters	Meaning	Defined on
n	total number of patients	–
m	total number of vehicles	–
$\bar{m}$	the set of all vehicles	–
T_V	maximum vehicle route duration	–
Q	vehicle capacity	–
d_{GP}	constant appointment duration	–
W	maximum patient time window	–
t_{ij}	travel time from i to j	$(i, j) \in N \times N$
d_i	service duration at i	$i \in N$
L_i	maximum user ride time	$i \in P$
e_i	earliest arrival time at i	$i \in N$
l_i	latest arrival time at i	$i \in N$
q_i	load of i	$i \in N$

Appendix B: Pseudocode MCLIH

Algorithm 1 gives an overview of the Mini-Cluster Linking Insertion Heuristic (MCLIH).

Algorithm 1 Mini-Cluster Linking Insertion Heuristic (MCLIH)

Input: Transportation requests $R = R_c \cup \overline{R_c} \cup R_w \cup \overline{R_w}$ with corresponding DARP road graph $G = (N, A)$, vehicle fleet size m, vehicle capacity Q, maximum route length T_V, maximum user ride time L_i for each request r_i, mini-clustering parameter ρ.
Compute set of mini-clusters $\mathcal{V}$ by using modified Kruskal's algorithm on auxiliary graph G'
for $\mathcal{M} \in \mathcal{V}$ **do**
    Compute intra-cluster route $\mathcal{S}$ on $\mathcal{M}$ by solving PDP formulation
end for
Construct complete directed auxiliary graph $H = (\mathcal{V}, \mathcal{A}, c)$ where $c(\mathcal{M}_i, \mathcal{M}_j)$ reflects distance between last stop of $\mathcal{S}_i$ and first stop of $\mathcal{S}_j$
Compute TSP tour T on H
Construct auxiliary graph $H' = (\mathcal{V}, E')$ modeling all feasible vehicle routes obtained by splitting T; see Labadie et al. (2016); Prins (2004)
Compute shortest path on H', obtaining vehicle schedules $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_m$
for $r \in R_c$ **do**
    Set appointment time after arrival of r
    Construct time windows for corresponding inbound request $\overline{r} \in \overline{R_c}$
    Insert $\overline{r}$ into $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_m$ using GIH (if it fails: remove and re-insert r and $\overline{r}$ with different time windows)
end for
Notify chronic patients about appointments, start handling walk-in rides on request
for $(r, \overline{r}) \in (R_w \cup \overline{R_w})$ **do**
    Insert r and $\overline{r}$ into $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_m$ by GIH in an online fashion
    if Insertion successful **then**
        continue
    else
        Reject $(r, \overline{r})$
    end if
end for
return $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_m$

Appendix C: Static DARP MIP

The static DARP can be formulated as the following Mixed Integer Program cf. Cordeau (2006). As explained in Cordeau (2006), this formulation can be strengthened via cutting planes if $Q_k = Q$.

$$\min \quad \sum_{i\in N}\sum_{j\in N}\sum_{k\in\bar{m}} c_{ij}^k x_{ij}^k$$

$$\text{s.t.} \quad \sum_{k\in\bar{m}}\sum_{j\in N} x_{ij}^k = 1 \qquad \forall i \in P \quad \text{(All Pickups take place)}$$

$$[1ex]\sum_{j\in N} x_{ij}^k - \sum_{j\in N} x_{n+i,j}^k = 0 \qquad i \in P,\, k \in \bar{m} \quad \text{(Started jobs are finished)}$$

$$[1ex]\sum_{j\in N} x_{0j}^k = 1 \qquad \forall k \in \bar{m} \quad \text{(Vehicles start at depot)}$$

$$[1ex]\sum_{i\in N} x_{i,2n+1}^k = 1 \qquad \forall k \in \bar{m} \quad \text{(Vehicles end at depot)}$$

$$[1ex]\sum_{j\in N} x_{ji}^k - \sum_{j\in N} x_{ij}^k = 0 \qquad \forall i \in P \cup D,\, k \in \bar{m} \quad \text{(Flow conservation)}$$

$$[1ex]B_i^k + d_i + t_{ij} - M_{ij}^k(1 - x_{ij}^k) \le B_j^k \qquad \forall i \in N,\, j \in N,\, k \in \bar{m} \quad \text{(Service times)}$$

$$[1ex]Q_i^k + q_j - W_{ij}^k(1 - x_{ij}^k) \le Q_j^k \qquad \forall i \in N,\, j \in N,\, k \in \bar{m} \quad \text{(Load)}$$

$$[1ex]B_{n+i}^k - \left(B_i^k + d_i\right) \le L_i \qquad \forall i \in P,\, k \in \bar{m} \quad \text{(User ride time)}$$

$$[1ex]B_{2n+1}^k - B_0^k \le T_V \qquad \forall k \in \bar{m} \quad \text{(Route length)}$$

Where the variables are given by

$$B_i^k \in [e_i, l_i] \qquad \forall i \in N,\, k \in \bar{m} \qquad \text{(Arrival times)}$$

$$[1ex]Q_i^k \in \left[\max\{0, q_i\},\, \min\{Q_k,\, Q_k + q_i\}\right] \qquad \forall i \in N,\, k \in \bar{m} \qquad \text{(Loads)}$$

$$[1ex]x_{ij}^k \in \{0, 1\} \qquad \forall i \in N,\, j \in N,\, k \in \bar{m} \qquad \text{(Route)}$$

Appendix D: Extended DARPCF MIP

Note that, apart from introducing the different patient and journey types, the most important difference is that the maximization of scheduled rides constitutes the new objective function.

$$
\begin{aligned}
\max \quad & \sum_{i\in P^*}\sum_{j\in N}\sum_{k\in\bar{m}} x_{ij}^k + \sum_{i\in P}\sum_{j\in N}\sum_{k\in\bar{m}} x_{ij}^k \\
\text{s.t.} \quad & \sum_{j\in N} x_{ij}^k - \sum_{j\in N} x_{n+i,j}^k = 0 && i \in P \cup \overline{P}, k \in \bar{m} \quad \text{(Started jobs are finished)} \\
& \sum_{j\in N} x_{0j}^k = 1 && \forall k \in \bar{m} \quad \text{(Vehicles start at depot)} \\
& \sum_{i\in N} x_{i,2n+1}^k = 1 && \forall k \in \bar{m} \quad \text{(Vehicles end at depot)} \\
& \sum_{j\in N} x_{ji}^k - \sum_{j\in N} x_{ij}^k = 0 && \forall i \in N \setminus \{0, 2n+1\}, k \in \bar{m} \quad \text{(Flow conservation)} \\
& B_i^k + d_i + t_{ij} - M_{ij}^k(1 - x_{ij}^k) \le B_j^k && \forall i \in N, j \in N, k \in \bar{m} \quad \text{(Service times)} \\
& Q_i^k + q_j - W_{ij}^k(1 - x_{ij}^k) \le Q_j^k && \forall i \in N, j \in N, k \in \bar{m} \quad \text{(Load)} \\
& B_{n+i}^k - (B_i^k + d_i) \le L_i && \forall i \in P \cup \overline{P}, k \in \bar{m} \quad \text{(User ride time)} \\
& B_{2n+1}^k - B_0^k \le T_V && \forall k \in \bar{m} \quad \text{(Route length)} \\
& \sum_{j\in N}\sum_{k\in\bar{m}} x_{ij}^k - \sum_{j\in N}\sum_{k\in\bar{m}} x_{i,j}^k = 0 && i \in P \quad \text{(Out- and inbound link)} \\
& B_{i+n}^k + d_{GP} \le B_{\bar{i}}^k && \forall i \in P, k \in \bar{m} \quad \text{(Appointment duration)} \\
& B_{\bar{i}}^k \le B_{i+n}^k + d_{GP} + W && \forall i \in P, k \in \bar{m} \quad \text{(Appointment duration)}
\end{aligned}
$$

Where the variables are given by

$$
\begin{aligned}
B_i^k &\in [e_i, l_i] && \forall i \in N, k \in \bar{m} \quad \text{(Arrival times)} \\
Q_i^k &\in [\max\{0, q_i\}, \min\{Q_k, Q_k + q_i\}] && \forall i \in N, k \in \bar{m} \quad \text{(Loads)} \\
x_{ij}^k &\in \{0, 1\} && \forall i \in N, j \in N, k \in \bar{m} \quad \text{(Route)}
\end{aligned}
$$

Acknowledgements The authors thank the anonymous reviewers for numerous helpful suggestions that improved the manuscript.

Author's contribution FR developed the algorithms, implemented them, and evaluated results. Furthermore, he wrote the paper. EA implemented the integer program. CB acquired funding and revised the text. MC generated the testing data, co-developed the algorithms and wrote the paper. FE generated the testing data and revised the text.

Funding Open Access funding enabled and organized by Projekt DEAL. This work was supported by the Freigeist-Fellowship of the Volkswagen Stiftung; the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 443158418; GRK2236/2 – 282652900; the German Federal Ministry of Education and Research (grant no. 05M16PAA) within the project "HealthFaCT-Health: Facility Location, Covering and Transport"; and by the special research fund of KU Leuven (project C14/22/026).

Data availability All data is publicly available or derived from the SiM-Care simulation.

Code availability Code can be made available upon request.

Declarations

Conflict of interest Not applicable.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

Adelhütte D, Braun K, Liers F, Tschuppik S (2021) Minimizing delays of patient transports with incomplete information including covid-19 requirements. http://www.optimization-online.org/DB_HTML/2021/02/8242.html

Ahern A, Hine J (2012) Rural transport-Valuing the mobility of older people. Res Transp Econ 34(1):27–34. https://doi.org/10.1016/j.retrec.2011.12.004

Beasley J (1983) Route first-cluster second methods for vehicle routing. Omega 11(4):403–408. https://doi.org/10.1016/0305-0483(83)90033-6

Bent RW, Van Hentenryck P (2004) Scenario-based planning for partially dynamic vehicle routing with stochastic customers. Oper Res 52(6):977–987. https://doi.org/10.1287/opre.1040.0124

Berg J, Ihlström J (2019) The importance of public transport for mobility and everyday activities among rural residents. Soc Sci 8(2):58. https://doi.org/10.3390/socsci8020058

Bodin LD, Sexton T (1986) The multi-vehicle subscriber dial-a-ride problem. TIMS Stud Manage Sci 26:73–86

Büsing C, Comis M, Rauh F (2021). The dial-a-ride problem in primary care with flexible scheduling. (arXiv:2105.14472) 10.48550/arXiv.2105.14472 arxiv:2105.14472 [math]

Cappanera P, Scutellà MG (2015) Joint assignment, scheduling, and routing models to home care optimization: a pattern-based approach. Transp Sci 49(4):830–852. https://doi.org/10.1287/trsc.2014.0548

Census 2011. RDC of the federal statistical office and statistical offices of the federal states of Germany (2011). https://doi.org/10.21242/12111.2011.00.04.1.1.0

Census 2022. RDC of the Federal Statistical Office and Statistical Offices of the Federal States of Germany (2022)

Christiaens J, Vanden Berghe G (2020) Slack induction by string removals for vehicle routing problems. Transp Sci 54:417–433. https://doi.org/10.1287/trsc.2019.0914

Comis M, Cleophas C, Büsing C (2021). Patients, primary care, and policy: agent-based simulation modeling for health care decision support. Health Care Manage Sci, pp 1–28. https://doi.org/10.1007/s10729-021-09556-2

Cordeau J-F, Iori M, Vezzali D (2024) An updated survey of attended home delivery and service problems with a focus on applications. Ann Oper Res 343:885–922. https://doi.org/10.1007/s10479-024-06241-9

Cordeau J-F (2006) A branch-and-cut algorithm for the dial-a-ride problem. Oper Res 54(3):573–586. https://doi.org/10.1287/opre.1060.0283

Cordeau J-F, Laporte G (2003) A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transp Res Part B Methodol 37(6):579–594. https://doi.org/10.1016/s0191-2615(02)00045-0

Cordeau JF, Iori M, Vezzali D (2023) A survey of attended home delivery and service problems with a focus on applications. 4OR 21(4):547–583. https://doi.org/10.1007/s10288-023-00556-2

Curtois T, Landa-Silva D, Qu Y, Laesanklang W (2018) Large neighbourhood search with adaptive guided ejection search for the pickup and delivery problem with time windows. EURO J Transp Logist 7(2):151–192. https://doi.org/10.1007/s13676-017-0115-6

Desrosiers J, Dumas Y, Soumis F (1988) The Multiple Vehicle DIAL-A-RIDE Problem. In: Daduna JR, Wren A (eds) Computer-aided transit scheduling, pp 15–27. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-85966-3_3

Dodd L (2013) Java implementation of ant swarm TSP solver. GitHub. https://github.com/lukedodd/ant-tsp

Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybernet Part B (Cybernetics) 26(1):29–41. https://doi.org/10.1109/3477.484436

Esmer B.C (2017). Java program for Kruskal's algorithm. GitHub. https://github.com/barisesmer/Algorithm-Bundle

Fikar C, Hirsch P (2017) Home health care routing and scheduling: a review. Comput Oper Res 77:86–95. https://doi.org/10.1016/j.cor.2016.07.019

Gaul D, Klamroth K, Stiglmayr M (2021). Solving the dynamic dial-a-ride problem using a rolling-horizon event-based graph. In: Müller-Hannemann M, Perea F (eds) 21st Symposium on algorithmic approaches for transportation modelling, optimization, and systems (ATMOS 2021). Open Access Series in Informatics (OASIcs), vol 96, pp 8–1816. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany. https://doi.org/10.4230/OASIcs.ATMOS.2021.8. https://drops.dagstuhl.de/opus/volltexte/2021/14877

Glover F (1996). Ejection chains, reference structures and alternating path methods for traveling salesman problems. Disc Appl Math 65(1):223–253. https://doi.org/10.1016/0166-218X(94)00037-E. First International Colloquium on Graphs and Optimization

Grenouilleau F, Legrain A, Lahrichi N, Rousseau LM (2019) A set partitioning heuristic for the home health care routing and scheduling problem. Eur J Oper Res 275(1):295–303. https://doi.org/10.1016/j.ejor.2018.11.025

Gschwind T, Drexl M (2019) Adaptive large neighborhood search with a constant-time feasibility test for the dial-a-ride problem. Transp Sci 53(2):480–491. https://doi.org/10.1287/trsc.2018.0837

Gschwind T, Irnich S (2015) Effective handling of dynamic time windows and its application to solving the dial-a-ride problem. Transp Sci 49(2):335–354. https://doi.org/10.1287/trsc.2014.0531

Gupta D, Denton B (2008) Appointment scheduling in health care: challenges and opportunities. IIE Trans actions 40(9):800–819. https://doi.org/10.1080/07408170802165880

Gurobi Optimization, LLC: gurobi optimizer reference manual. (2023) https://www.gurobi.com

Ho S.C, Szeto W.Y, Kuo Y.-H, Leung J.M.Y, Petering M, Tou T.W.H (2018). A survey of dial-a-ride problems: Literature review and recent developments. Transp Res Part B Methodol 111:395–421. https://doi.org/10.1016/j.trb.2018.02.001

Ioachim I, Desrosiers J, Dumas Y, Solomon MM, Villeneuve D (1995) A request clustering algorithm for door-to-door handicapped transportation. Transp Sci 29(1):63–78. https://doi.org/10.1287/trsc.29.1.63

Jaw J-J, Odoni AR, Psaraftis HN, Wilson NHM (1986) A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. Transp Res Part B Methodol 20(3):243–257. https://doi.org/10.1016/0191-2615(86)90020-2

Johnn S-N, Zhu Y, Miniguano-Trujillo, A, Gupte A (2021) The home service assignment, routing, and appointment scheduling (H-SARA) problem with uncertainties*. https://doi.org/10.13140/RG.2.2.16295.88485

Kalantari B, Hill AV, Arora SR (1985) An algorithm for the traveling salesman problem with pickup and delivery customers. Eur J Oper Res 22(3):377–386. https://doi.org/10.1016/0377-2217(85)90257-7

Labadie N, Prins C, Prodhon C (2016) metaheuristics for vehicle routing problems. John Wiley & Sons, Inc., Hoboken, NJ, USA. https://doi.org/10.1002/9781119136767

Luxen D, Vetter C (2011) Real-time routing with OpenStreetMap data. In: Proceedings of the 19th ACM SIGSPATIAL international conference on advances in geographic information systems. GIS '11, pp 513–516. ACM, New York, NY, USA. https://doi.org/10.1145/2093973.2094062

Madsen OBG, Ravn HF, Rygaard JM (1995) A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. Ann Oper Res 60(1):193–208. https://doi.org/10.1007/BF02031946

Mladenović N, Hansen P (1997) Variable neighborhood search. Comput Oper Res 24(11):1097–1100. https://doi.org/10.1016/S0305-0548(97)00031-2

Molenbruch Y, Braekers K, Caris A (2017) Typology and literature review for dial-a-ride problems. Ann Oper Res 259(1–2):295–325. https://doi.org/10.1007/s10479-017-2525-0

Parragh SN, Doerner KF, Hartl RF (2008) A survey on pickup and delivery problems: part II: transportation between pickup and delivery locations. Journal für Betriebswirtschaft 58(2):81–117. https://doi.org/10.1007/s11301-008-0036-4

Prins C (2004) A simple and effective evolutionary algorithm for the vehicle routing problem. Compu Oper Res 31(12):1985–2002. https://doi.org/10.1016/S0305-0548(03)00158-8

Prins C, Lacomme P, Prodhon C (2014) Order-first split-second methods for vehicle routing problems: a review. Transp Res Part C Emerg Technol 40:179–200. https://doi.org/10.1016/j.trc.2014.01.011

Psaraftis HN (1980) A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. Transp Sci 14(2):130–154. https://doi.org/10.1287/trsc.14.2.130

Schilde M, Doerner KF, Hartl RF (2011) Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. Comput Oper Res 38(12):1719–1730. https://doi.org/10.1016/j.cor.2011.02.006

Syed ST, Gerber BS, Sharp LK (2013) Traveling towards disease: transportation barriers to health care access. J Commun Health 38(5):976–993. https://doi.org/10.1007/s10900-013-9681-1

Wilson NH, Sussman JM, Wong HK, Higonnet T (1971). Scheduling algorithms for a dial-a-ride system. Massachusetts Institute of Technology. Urban Systems Laboratory Report

Authors and Affiliations

Felix Rauh[1,2] · Emma Ahrens[1] · Christina Büsing[1] · Martin Comis[1] · Felix Engelhardt[1]

✉ Felix Rauh
felix.rauh@rwth-aachen.de

✉ Felix Engelhardt
engelhardt@combi.rwth-aachen.de

Emma Ahrens
ahrens@cs.rwth-aachen.de

Christina Büsing
buesing@combi.rwth-aachen.de

Martin Comis
comis@math2.rwth-aachen.de

[1] Teaching and Research Area Combinatorial Optimization, RWTH Aachen University, Templergraben 55, 52062 Aachen, Germany

[2] Research Center for Operations Management, KU Leuven, Naamsestraat 69, 3000 Leuven, Belgium