Optimization Letters (2022) 16:355–374
https://doi.org/10.1007/s11590-020-01698-6

ORIGINAL PAPER

Bi-objective optimization model for the heterogeneous dynamic dial-a-ride problem with no rejects

André L. S. Souza[1] · Marcella Bernardo[2] · Puca H. V. Penna[1] · Jürgen Pannek[3] · Marcone J. F. Souza[1]

Received: 31 March 2020 / Accepted: 28 December 2020 / Published online: 8 January 2021

Abstract
This work proposes a bi-objective mathematical optimization model and a two-stage heuristic for a real-world application of the heterogeneous Dynamic Dial-a-Ride Problem with no rejects, i.e., a patient transportation system. The problem consists of calculating route plans to meet a set of transportation requests by using a given heterogeneous vehicle fleet. These transportation requests can be either static or dynamic, and all of them must be attended to. In the first stage of the proposed heuristic, the problem's static part is solved by applying a General Variable neighborhood Search based algorithm. In the second stage, the dynamic requests are dealt with by implementing a simple insertion heuristic. We create different instances based on the real data provided by a Brazilian city's public health care system and test the proposed approach on them. The analysis of the results shows that the higher the level of dynamism, i.e., the number of urgent requests on each instance, the smaller the objective function value will be in the static part. The results also demonstrate that a higher level of dynamism increases the chance of a time window violation happening. Besides, we use the weighted sum method of the two conflicting objectives to analyze the trade-off between them and create an approximation for the *Pareto* frontier.

Keywords Dial-a-ride problem · Vehicle routing · Dynamic DARP · General variable neighborhood search

1 Introduction

In this work, we study a transportation problem that arises in the daily operations of a municipal health care system in Brazil: the transportation of patients. Transportation of patients is an essential service worldwide, especially in Brazil, where some health treatments and services are offered only by hospitals located in large cities. In Brazil, not only medical services but also the transportation of patients is provided by the municipality. In the city of Ouro Preto, the transportation of patients occurs between a

Extended author information available on the last page of the article

patient's home or a public location in the city and hospitals in the metropolitan region of Belo Horizonte and Itabirito (or vice-versa). The local public health system classifies the patients' transportation requests according to the treatment and/or service prescribed by the patient's doctor. If the patient needs immediate treatment/service, such as surgery, the request is classified as urgent, and the patient must then be attended on the same day (dynamic request). The request can otherwise be attended in the next days or weeks (static request). To deal with such situation, the health system has a heterogeneous fleet of vehicles. Not all vehicles are allocated each day. Some of them are saved to be used for attending a possible urgent transportation request. Nevertheless, when there is no urgent transportation request, the same amount of vehicles are idle and under-utilised. The public system also recognised long waiting times when an urgent request could not be attended to, increasing the patient inconvenience. In this way, this work's main goal is to develop a mathematical model to improve the transportation service provided by the municipal health care system of Ouro Preto.

The problem faced by the local health system can be classified as a variant of the Vehicle Routing Problem (VRP). The VRP goal is to determine a set of routes to be executed by a fleet of vehicles for serving geographically dispersed customers [2]. Several VRP variants include real-life complexities. Among this variety of classes is the Dial-a-Ride Problem (DARP). The main difference between the DARP and other VRP variants is the human perspective. In other words, the DARP considers the convenience and comfort of customers. We can find several DARP classifications in the literature. The best-known classification is based on the type of requests, and it distinguishes the problem into two classes: static and dynamic. In the Static DARP (SDARP), all problem inputs are known a-priori, i.e., before the construction of the routes, whereas in the Dynamic DARP (DDARP), the route plan is updated every time new information is revealed.

The DDARP has received less attention than the SDARP. The transportation of patients (health care) is the major application of the DDARP [9]. Madsen et al. [12] studied the multi-objective heterogeneous DDARP to tackle the problem faced by the Copenhagen Fire-Fighting Service (CFFS) in transporting elderly and disabled persons. Their objective function includes several goals, which were weighted by parameters to reflect the decision-maker's preferences. For solving the problem, the authors developed an insertion algorithm. Hanne et al. [6] also studied the multi-objective heterogeneous DDARP in which the overall objective is to minimize a weighted sum of four objectives. Nevertheless, in Madsen et al. [12] dynamism refers to the occurrence of a new request, whereas Hanne et al. [6] considered not only new requests but also vehicle-related events (e.g., vehicle breakdown) and availability of staff member as dynamic inputs. The authors developed a software for managing all the activities related to patient transportation of a German hospital, namely travel booking, scheduling, dispatching, monitoring, and reporting transports. The proposed software includes a discrete-event simulation model to reproduce the behavior of the patient transportation system and several optimization routines that can be combined depending on the time available for planning. Similarly to Hanne et al. [6], Beaudry et al. [1] adopted new requests and vehicle breakdowns as sources of dynamism in the heterogeneous DDARP to solve the problem arising in the daily operation of a German hospital. The proposed model includes a bi-objective function (minimization of fleet operating costs and the maximisation of patient satisfaction). The authors developed a two-phase heuristic, which includes an insertion scheme (first phase) and a tabu search algorithm (second phase). The work of Schilde et al. [14] differs from the above papers concerning stochasticity and type of vehicle fleet. In the previous papers, stochastic information about the dynamic inputs was not available, while Schilde et al. [14] used historical data about some transportation requests to exploit stochastic information. They called the problem the dynamic stochastic dial-a-ride problem with expected return transports. The paper's goal was to investigate whether using stochastic information for calculating route plans positively affects the quality of the solution. To reflect the problem faced by the Austrian Red Cross, the authors assumed that patients occupy only one seat in a vehicle of the homogenous fleet. To solve the problem, Schilde et al. [14] adopted four well-establish metaheuristics, which were modified to consider stochastic information.

In this work, we formulated the problem faced by Ouro Preto's health care system as a bi-objective heterogeneous DDARP with no rejects. In this problem, urgent transportation requests are the source of dynamism, and all dynamic and static transportation requests must be served. The conflicting objectives are the minimization of the transportation costs and minimization of user inconvenience. We propose a two-stage heuristic to solve the problem. The first stage solves the static requests, designing the so-called a-priori route plan. For that, we use an algorithm based on the General Variable Neighborhood Search (GVNS) [8]. This GVNS-based algorithm applies a variant of the Variable Neighborhood Descent (VND) as local search method. GVNS is an algorithm that has successfully solved several other combinatorial problems, e.g., drone delivery [5], location routing [10], pollution routing [11], open-pit mining [15], among others. In the second stage, for dealing with the dynamic transportation requests, an insertion heuristic is applied. The main difference between our work and the above-mentioned papers is that 1) we analyse the impact of the degree of dynamism of the transportation requests on the quality of the solutions and 2) we create an efficient frontier, i.e., an approximation to the *Pareto* frontier, by trading off parameter α included in the bi-objective function. Therefore, this paper's contributions are threefold: a new two-stage heuristic for the DDARP and dynamism and *Pareto* analyses. The remainder of the paper is organised as follows. In Sect. 2, the real transportation problem is presented. This is followed in Sect. 3 by a description of the problem formulation. We devote Sect. 4 to the proposed solution method and definition of dynamism in the problem. The dataset created for testing our approach is presented in Sect. 5. Dynamism and trade-off analyses are presented in the computational experiments presented in Sect. 6. Lastly, Sect. 7 concludes with a summary and an outlook for future work.

2 Problem description

The problem considered in this work is based on a real problem faced by the public health care system of Ouro Preto, Brazil. The problem is explained as follows. At the beginning of a workday, the health care system's transportation department schedules transportation requests and calculates a route plan to be executed during that

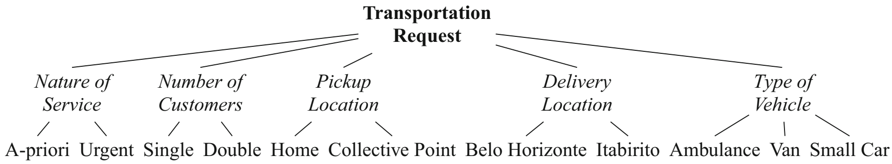


Fig. 1 Characteristics of the problem

day. For this, only the requests known a-priori are considered. Nonetheless, urgent transportation request might occur throughout the day. The main characteristics of a transportation request are presented in Fig. 1 and explained bellow.

An transportation request is classified as either *a-priori* known or *urgent*. A transportation request implies either one or two passengers, i.e., either only a *single* patient or one patient and one companion person (*double*) when the former is unable to travel alone. If the *double* customer is a stretcher patient, the patient requires being transported in an ambulance together with the companion person. A transportation request can have either of the two types of pickup locations: *home* or *collective points*. Urgent patients and patients who cannot move have to be collected at their homes; otherwise, they are picked up at collective points. These collective points are distributed around the city and are characterised as something representative or important in the city, such as big squares, old churches, and train station. The collective point in which the patient has to be collected at is selected following the shortest distance between the patient's home and the collective point. A transportation request can have either Belo Horizonte or Itabirito as delivery locations.

Figure 2 displays both pickup and delivery areas. The area contained in the blue circle comprises all the possible pickup points. In contrast, the areas contained in the two red circles include all the possible delivery points, where hospitals and health clinics are located. Since the treatments/services offered by hospitals located in large cities have time-restricted work, the time windows are limited at delivery points. Finally, regarding the type of vehicles, the transportation sector has a heterogeneous fleet of vehicles. This fleet is composed by *ambulances*, *vans*, and *small cars*. Therefore, each vehicle has a different capacity and a cost (renting cost) associated with it. Table 1 presents the characteristics of each vehicle. They are classified as A to E.

3 Mathematical formulation

The heterogeneous SDARP can be represented on a fully connected undirected graph $G = (V, A)$, where $V = (P \cup D)$ is the set of vertices, which denote patient locations (v_0 and v_{2n+1} represent the depot), and A is the set of edges, with $|A| = a$. The set V is subdivided into two sets, where $P = \{1, \dots, n\}$ and $D = \{n+1, \dots, 2n\}$ are the sets of pickup and delivery locations, respectively. Then, $|V| = 2n$. Cost c_{ij} and travel time δ_{ij} are fixed for all $(i, j) \in A$. The fleet of vehicles is represented by K, and for each vehicle $k \in K$ a capacity w_k has to be respected. Each patient $i \in P$ has a pickup i and delivery $n+i$ location, a load q_i, a non-negative service time τ_i, and a time window $[e_i, l_i]$ (where e_i and l_i are integers non-negative). A solution y, called

Fig. 2 Pickup and delivery areas

route plan, is a set of routes. Each route is done by one vehicle which leaves the depot, serves a subset of users whose total load and total travel time do not exceed Q_k and T_k, respectively, and returns to the depot in the interval $[e_0, l_0]$. A customer must first be picked up and then delivered by the same vehicle. For every customer, the service should start between $[e_i, l_i]$. That is, let Ar_i^k denote the arrival time of vehicle k at the vertex i, the beginning of the service (B_i^k) cannot start before the beginning of the time window ($B_i^k \geq \max\{Ar_i^k, e_i\}$). The departure time $D_i^k = B_i^k + \tau_i$ is the time vehicle k departures from customer i. Vehicle k waiting time is defined by $W_i^k = B_i^k - Ar_i^k$. The total duration of the route is calculated as $T_k = B_{n+1}^k - B_0^k$, where B_{n+1}^k and B_0^k represent when vehicle k finishes and starts its ride at the depot, respectively. x_{ij}^k is a binary variable that takes value 1 if edge (i, j) is used by vehicle k or zero otherwise. w_i^k is an integer variable that takes the number of people in vehicle k after visiting node i or zero if the vehicle does not serve the customer i. In this work, we adapt the formulation of the heterogeneous SDARP proposed by Cordeau [4]. Cordeau's SDARP mathematical model solves to optimality small and medium-size instances, and it has been studied extensively by other authors. In Cordeau's formulation, the objective function only minimizes the total routing cost. In contrast, we adopt a bi-objective function in which the first objective is to minimize transportation costs, and the second is to minimize user inconvenience. Thus, our model distinguishes from Cordeau's model in that the former introduces violations of time windows and ride time of each user as soft constraints to reject no request. Besides, when adding a parcel

Table 1 Vehicle information

Vehicle	Representation	Capacity	Cost	Number of users by type Non-stretcher	Stretcher	Companion
Small Car 1	A	4	100	2	0	2
Small Car 2	B	5	120	3	0	2
Ambulance 1	C	3	100	0	1	2
Ambulance 2	D	4	150	0	2	2
Van	E	16	200	8	0	8

to deal with user inconvenience in the objective function, we introduce n new variables, where n is the number of pickup requests. Then, the total number of variables in the proposed formulation is $(2n)^2 \times |K| + n$. The formulation of the SDARP is represented as follows.

$$\min \sum_{k\in K}\sum_{i\in V}\sum_{j\in V} c_{ij}^k x_{ij}^k \tag{1}$$

$$\min \frac{\sum_{p\in P} Los_p}{n} \tag{2}$$

$$\text{s.t.} \sum_{k\in K}\sum_{j\in V} x_{ij}^k = 1, \qquad \forall i \in P, \tag{3}$$

$$\sum_{k\in K}\sum_{j\in V} x_{ij}^k = 1, \qquad \forall i \in P, \tag{4}$$

$$\sum_{j\in V} x_{0j}^k = 1, \qquad \forall k \in K, \tag{5}$$

$$\sum_{i\in V} x_{i,2n+1}^k = 1, \qquad \forall k \in K, \tag{6}$$

$$\sum_{j\in V} x_{ij}^k - \sum_{j\in V} x_{n+i,j}^k = 0, \qquad \forall i \in P, \forall k \in K, \tag{7}$$

$$\sum_{i\in V} x_{il}^k - \sum_{j\in V} x_{lj}^k = 0, \qquad \forall l \in V, \forall k \in K, \tag{8}$$

$$b_j^k \geq b_i^k + \tau_i + \delta_{ij} - M1\left(1 - x_{ij}^k\right), \qquad \forall i \in V, \forall j \in V, \forall k \in K, \tag{9}$$

$$w_j^k \geq w_i^k + q_i - M2\left(1 - x_{ij}^k\right), \qquad \forall i \in V, \forall j \in V, \forall k \in K, \tag{10}$$

$$b_{2n+1}^k - b_0^k \leq T_k, \qquad \forall k \in K, \tag{11}$$

$$e_i \leq b_i^k \leq l_i, \qquad \forall i \in V, k \in K, \tag{12}$$

$$\max\{0, q_i\} \leq w_i^k \leq \min\{w_k, w_k + q_i\}, \qquad \forall i \in V, k \in K, \tag{13}$$

$$x_{ij}^k \in \{0, 1\}, \qquad \forall i \in V, j \in V, k \in K. \tag{14}$$

Using the parameter $\alpha \in [0, 1]$, we combine the two objectives transportation costs and user inconvenience, given by Eqs. (1) and (2), respectively, into a mono-objective function (15), which represents the weighted sum of these two objectives.

$$\min\ J(y) := \left\{ \alpha \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} x_{ij}^{k} c_{ij}^{k} + (1-\alpha) \frac{\sum_{p \in P} Los_p}{n} \right\} \tag{15}$$

The user inconvenience for user $p \in P$ which is served by vehicle k is defined by the expression $Los_p = \frac{(Ar_p^k - B_p^k)}{L(v_p, v_{n+p}, B_p^k)}$, where $L\left(v_p, v_{n+p}, B_p^k\right)$ represents the shortest path between v_p and v_{n+p} beginning the service at p at time B_p^k. In Eq. (15), if α is chosen large, then the model favours transportation cost by trying to include as many as possible users in one route. On the other hand, if α is small, there will be user discontent since few users will be served on a route. Therefore, the weight α can be used by managers to prioritise the importance of each of the two objective functions. Constraints (3), (4), and (5) ensure that all customers are served only once and all routes start and end at the depot. Constraints (6) guarantees a customer is attended by the same vehicle. Constraints (7) ensure that each costumer's pickup and delivery will be done by the same vehicle. Constraints (8) and (9) state that the consistency of time variables and the vehicle capacity are respected, respectively, where $M1 \geq \max\{0, l_i + \tau_i + \delta_{ij} - e_j\}$ and $M2 \geq \min\{w_k, w_k + q_i\}$. Constraints (10) ensure that the vehicles comply with the defined capacity limit imposed. The starting time of the service at each pickup node is guaranteed by Constraints (11). The vehicles capacity is represented by Constraints (13). Finally, Constraints (14) set the domains of the binary decision variables.

4 Solution approach

Since the mathematical model described in the previous section cannot solve real instances of the dynamic DARP, we developed a heuristic algorithm that includes two stages. The first stage happens at the beginning of a workday and designs an a-priori route plan considering the patients' request known at that time, and the second stage occurs at run time and deals with the dynamic patients' requests. For each dynamic request, an occurrence time o_i is defined, and the insertion time of the dynamic request is determined by $o_i < e_i$. The two stages of the proposed algorithm are described in the following sections.

4.1 First stage

The a-priori solution is calculated by using a GVNS-based algorithm [8]. Algorithm 1 presents the pseudo-code of the implemented GVNS. The GVNS algorithm starts from an *initial solution* and requires the determination of three components: *stopping criterion*, *shaking* procedure, and *local search* procedure. For designing the *initial solution*, we use a greedy function that selects each request based on the smaller time window. The request with the smaller time window is allocated to the vehicle with traveled time closer to the request's time window. The *stopping criterion* is defined by the number of iterations without improvement in the current solution ($iterMax$ = 1000). The *shaking procedure* is applied using a set $\mathcal{N}'$ containing five neighborhoods based on classical inter-route operators ($\mathcal{N}'$ = {Shift(1,0), Shift(2,0), Swap(1,1), Swap(2,2), Swap(2,1)}, see [13]). If the number of iterations without improvement ($iter$) is smaller than 25% of the number of requests, the shaking procedure is applied $iter$ times, each time randomly selecting one of the neighborhoods. Thus, the shaking intensity increases as there is no improvement in the current solution. The *local search* procedure is made by the Randomized Variable Neighborhood Descent (RVND) method [13,15,16].

Algorithm 1: GVNS ($s, iterMax$)

```
s ← initialSolution();
N = {Relocation, Swap, Crossover}
N' = {Shift(1, 0), Shift(2, 0), Swap(1, 1), Swap(2, 1), Swap(2, 2)}
iter ← 0
while iter ≤ iterMax do
    s' ← Shaking(s, N', iter)
    s'' ← RVND(s', N)
    if f(s'') < f(s) then
        s ← s''
        iter ← 1
    else
        iter ← iter + 1
    end
end
return s
```

RVND is a variant of the Variable Neighborhood Descent (VND) [7] procedure with random neighborhood ordering. It has the following differences concerning the Basic VND, also known as B-VND [8]. Firstly, we randomly select a neighborhood, that is, there is no prefixed neighborhood order. Secondly, whenever there is an improvement in the current solution, we rebuild the set of neighborhoods, and the search restarts from a randomly selected neighborhood. Like the B-VND procedure, RVND ends when the current solution cannot be improved concerning all the neighborhoods used. According to [16], RVND avoids looking for the best order, which may be highly dependent on the instance. Algorithm 2 shows the pseudo-code of RVND. It works in the following way. Initially, a neighborhood $\mathcal{N}_k$ is chosen randomly from the set $\mathcal{N}$ of neighborhoods (line 2). Then, the current solution s is submitted to a local search in this neighborhood $\mathcal{N}_k$ with the first improvement strategy (line 3). We follow the first improvement search strategy because this strategy requires lower computational time than the best improvement strategy while maintaining the same solution's quality compared to the best improvement strategy. If the resulting solution s' from the local search is better than the current solution s, then s is updated, and a new order of neighborhoods is established (lines 5-6). Otherwise, we remove the neighborhood $\mathcal{N}_k$ from $\mathcal{N}$.

Algorithm 2: RVND(s, $\mathcal{N}$)

1 **while** ($\mathcal{N} \neq \emptyset$) **do**
2     $\mathcal{N}_k \leftarrow$ random neighborhood of $\mathcal{N}$
3     $s' \leftarrow \mathit{FirstImprovement}(s, \mathcal{N}_k)$
4   **if** $f(s') < f(s)$ **then**
5         $s \leftarrow s'$
6   **rebuild** $\mathcal{N}$
7   **else**
8         $\mathcal{N} \leftarrow \mathcal{N} \setminus \{\mathcal{N}_k\}$
9   **end**
10 **end**
11 **return** s

4.2 Second stage

In the second stage, we use an insertion heuristic based on the heuristic proposed by Campbell and Savelsbergh [3]. This insertion heuristic works as follows. When one dynamic request arrives at time $T = o_i$, the closest available vehicle is allocated to this request. While the transportation requests served before $T = o_i$ cannot be changed, the others can. This means that the remaining sequence of customers to be attended according to the a-priori route plan at $T = o_i$ may be modified if savings can be achieved. Although the emergency request has priority in the allocated vehicle, any request in this vehicle that has a delivery point between the pickup and delivery location of the emergency request should be delivered before the urgent request. For instance, an emergency patient is allocated to car z, and the pickup and delivery location of this patient are points A and C. One of the customers in car z has to be dropped off at hospital B. This hospital is situated between A and C, this patient will thus be delivered before the emergency patient. Figure 3 illustrates the situation described above. In this figure, the black dots represent the pickup points - either home or collective point - and the gray dots represent the delivery locations. At the beginning of the time horizon ($T = 0$) the a-priori route plan calculated by the proposed GVNS is shown in dashed line. After that, the urgent request appears at $T = o_i$. The part of the a-priori route plan that has already been executed by the vehicle z when the urgent request occurs is shown as a continuous line. The emergency pickup A and delivery C points are displayed in orange. After the emergency request is delivered, the remaining route is recalculated by prioritizing a shorter time window. The final route plan, which is the route plan after the execution of all transportation requests, is exhibited at $T = T_f$.

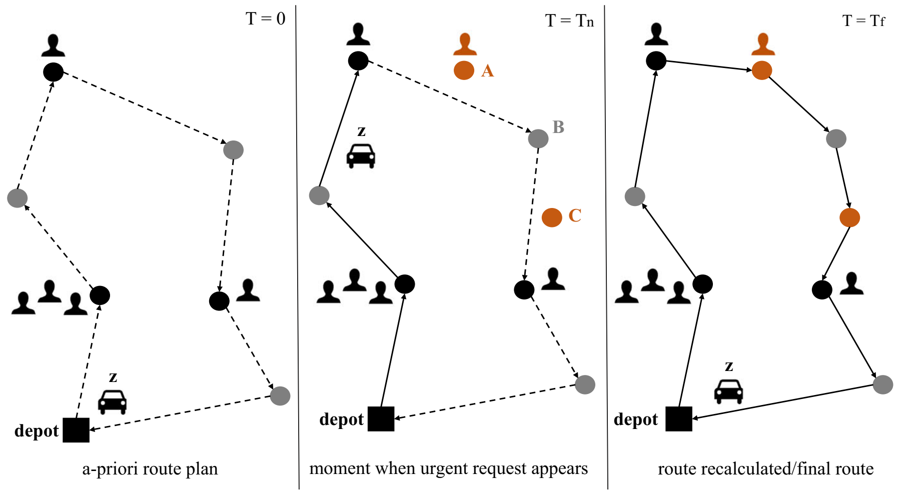


Fig. 3 Inclusion of emergency request in a route

5 Dataset

Since we want to tackle the real transportation problem faced by the health system of Ouro Preto, we developed a set of test problems based on real data to evaluate the proposed solution approach. The provided data show a schedule of some requests served by the transportation sector. These requests are related to non-sequential workdays that occurred between March and October 2019. Each transportation request includes the nature of service, the number of customers, pickup and delivery locations (see Fig. 1), and time windows.

We create four instances to test our proposal. Two instances, named a9-21 and a9-24, were designed based on information from two working days. Instance a9-21 has 21 transportation requests, and out of these 15 are single patients and 6 are double patients (patients with a companion). Instance a9-24 has 24 transportation requests, and out of these 11 are single patients, and 13 are double patients. The third instance, called a9-34, was created by joining four working days, computing 34 requests (12 single patients plus 22 double patients). The largest instance with 48 requests, named a9-48, was designed to simulate a really busy working day and contains 17 single patients and 31 double patients. To create Instance a9-48, we assume that half of the pickup locations were patients' homes, and the other half were collective points. To generate these homes' location, we randomly selected points in the area of the city (area contained in the blue circle in Fig. 2). To generate each one of the collective points' locations, we randomly selected one of the six collective points provided by the transportation sector.

Table 2 Results obtained with the proposed algorithm for Scenario 1

Instance	*Sol*		*RC*		*VC*		*Vehicle*	*qV*		
	Best	*Avg*	*Best*	*Avg*	*Best*	*Avg*		*Best*	*Avg*	*Time (s)*
a9-21	1019.51	1072.49	469.51	535.49	550	537	B_1, C_1, D_1, E_1	4	4	5.87
a9-24	1815.07	1820.61	845.07	850.61	970	970	A_4, B_1, C_1, D_1, E_1	8	8	8.77
a9-34	1443.31	1522.96	773.31	835.96	670	687	A_2, B_1, D_1, E_1	5	5.4	43.97
a9-48	2037.60	2088.84	1167.60	1228.84	870	860	A_3, B_1, C_1, D_1, E_1	7	7	147.08

6 Computational experiments

In this section, we investigate the impact that the level of dynamism has on the solutions' quality. For that, we created two scenarios: Scenario 1 and Scenario 2. These scenarios and the results obtained for each scenario are described in the following. Our implementations of the proposed solution method described in Sect. 4 were coded in C/C++. All experiments were performed on an Intel Core i7-5500U CPU @ 2.40 GHz × 4 with 8GB RAM, Ubuntu 16.04.5 LTS 64 bits. The code details and the designed instances are available at https://gitlab.com/AndreLuyde/dynamic-heterogeneous-darp/tree/master/Instances.

6.1 Scenario 1

In Scenario 1, we assumed that no urgent patient occurs in the problem, that is no dynamic request. That is to say that in Scenario 1 all transportation requests are beforehand known. Since there is no dynamic request, user inconvenience is not considered in this scenario, i.e., $\alpha = 1$ in Eq. (15). Therefore, in this scenario, the problem is treated as a static DARP.

We solved every instance 10 times, computing 10 solutions for each instance. Table 2 shows the results obtained by our solution approach considering Scenario 1. The second and third columns (*Sol*: *Best* and *Avg*) display the best and average values of the objective function, respectively. The objective function value is the sum of routing (*RC*) and vehicle (*VC*) costs, which have their best and average values shown from the fourth to the seventh columns. The column *Vehicle* shows which vehicles were used in the best solution (for information on vehicle types, see Table 1). The subscription of each letter represents the number of used vehicles of that type. The columns under *qV* display the number of vehicles used in the best solution and the average number of used vehicles, respectively. Finally, the last column presents the average time in seconds spent by our algorithm to calculate a route plan.

We noted that although Instance a9-24 is smaller than Instance a9-34, the solution calculated for Instance a9-24 had a higher objective function value and vehicle and routing costs than the route plan designed for Instance a9-34. This occurred because the time window of the transportation requests in Instance a9-24 is more distant than the ones in Instance a9-34. Thus, the route plan designed for Instance a9-24 required more vehicles and longer routes, increasing the objective function value. Actually, the route plan for Instance a9-24 presented the highest number of vehicles among all the solutions, followed by the solution for Instance a9-48. As expected, the solution calculated for Instance a9-48 had the highest objective function value and routing and vehicle costs. It can also be seen that the small vehicle B, ambulance D, and van E were used in all solutions since these vehicles can visit more patients at a lower cost than vehicle A. Lastly, regards to computational time, the execution time grew according to the size of the instances. Nevertheless, Instance a9-48 required an unexpected higher computational time.

6.2 Scenario 2

In Scenario 2, we adopted the occurrence of emergency requests. For constructing this scenario, we created a variable γ that exhibits the instance's dynamism. This dynamism displays the number of emergency requests that can happen during a day. Emergency requests can appear at any time of day, have priority to be served, and their pickup location is always the patient's home. We randomly generated γ so that every instance has up to 40% of dynamic requests, i.e., $\gamma \in [0; 40]$. Among the emergency requests, there are inactive requests, represented by ι. These requests are patients that may not appear in the system. For each instance this can be $\iota \in [0; 20]$. Since instances a9-21 and a9-24 have just a few pickup points, which are patient's home, the degree of dynamism for these instances is reduced.

In this scenario, it is meaningful to perform a trade-off analysis between the two conflicting objectives, minimization of the transportation cost and minimization of the patient inconvenience. A common procedure for bi-objective optimization problems is the weighted sum method. We discretised the parameter α of Eq. (15) in nine weight combinations, from 0.1 to 0.9. So, the weights α and $1-\alpha$ of the transportation cost and user inconvenience objective functions, respectively, vary as follows: $(\alpha, (1-\alpha)) \in \{(0.1, 0.9), (0.2, 0.8), \ldots, (0.9, 0.1)\}$. For each combination of weights, the Eq. (15) was solved by the proposed algorithm. We thus computed nine route plans for each instance, and every solution displayed an objective function value that balances transportation costs and user inconvenience. It is worth mentioning that to trade-off transportation cost and user inconvenience, we only considered the static transportation requests since the dynamic requests have priority, and their delivery is, therefore, done as soon as possible. In this way, the time windows become a flexible constraint, that is, a-priori known requests may arrive late at the delivery place.

Tables 3a–d present the results obtained for Scenario 2. In these tables, the first column indicates the weight α given to the first term of Eq. (15). The second column presents the degree γ of dynamism. The third column exhibits the objective function value, which is the sum of transportation cost (*TC*) plus user inconvenience (*Los*). The fourth column shows the transportation costs multiplied by the factor α. The fifth column shows the user inconvenience multiplied by $(1-\alpha)$. The column *Vehicle* displays the type and quantity of vehicles used in the solution. The column *Time (s)* presents the computational time, in seconds, spent by the solution approach to find each solution. Lastly, the column *DSCV* displays which solutions had time window violations after the dynamic requests were served.

Table 3 Results obtained with the proposed heuristic for Scenario 2

α	γ	*Sol*	$TC \times \alpha$	$Los \times (1-\alpha)$	Vehicles	Time (s)	DSCV[a]
(*a*) *Instance* a9-21							
0.9	0.1	104.14	99.33	4.81	A_4	7.13	
0.8	0.1	207.29	202.82	4.47	A_2, B_1, E_1	8.82	
0.7	0.1	308.07	304.22	3.86	A_2, C_1, D_1	5.91	*
0.6	0.1	388.93	385.63	3.30	A_3, C_1	7.17	
0.5	0.1	468.61	465.91	2.69	A_2, B_1, C_1	12.57	
0.4	0.1	580.64	578.44	2.20	A_4	10.51	
0.3	0.1	688.98	687.41	1.58	A_2, B_1, D_1	7.52	
0.2	0.1	812.67	811.65	1.02	A_2, B_1, C_1	8.07	
0.1	0.1	913.21	912.66	0.55	A_2, C_1, D_1	5.51	*
0.9	0.2	101.59	96.79	4.79	A_2, C_1, D_1	3.89	*
0.8	0.2	190.38	186.00	4.38	A_1, B_1, C_1, E_1	7.70	
0.7	0.2	262.95	259.27	3.69	A_2, B_1, C_1	7.07	
0.6	0.2	428.85	425.69	3.16	A_2, C_1, E_1	5.23	
0.5	0.2	461.29	458.51	2.78	A_3, C_1	3.34	
0.4	0.2	582.44	580.21	2.22	A_2, C_1, D_1	4.21	*
0.3	0.2	606.53	604.95	1.58	A_2, B_1, C_1	6.16	
0.2	0.2	756.43	755.38	1.05	A_3, E_1	4.51	
0.1	0.2	828.47	827.94	0.53	A_2, B_1, C_1	7.59	
0.9	0.3	96.75	91.99	4.76	A_2, B_1, C_1	9.99	
0.8	0.3	197.84	193.58	4.26	A_2, C_1, D_1	7.55	*
0.7	0.3	286.95	283.27	3.69	A_2, C_1, E_1	8.66	
0.6	0.3	387.78	384.52	3.26	A_2, C_1, E_1	7.92	
0.5	0.3	461.29	458.51	2.78	A_3, C_1	5.59	
0.4	0.3	560.96	558.79	2.17	A_1, B_1, C_1, D_1	6.37	
0.3	0.3	680.57	678.95	1.61	A_1, B_1, C_1, D_1	6.88	*
0.2	0.3	750.73	749.62	1.11	A_2, B_1, C_1	4.59	
0.1	0.3	778.32	777.80	0.53	A_2, B_1, C_1	6.92	
0.9	0.4	96.75	91.99	4.76	A_2, B_1, C_1	9.88	
0.8	0.4	204.22	199.99	4.23	A_2, C_1, E_1	5.66	
0.7	0.4	268.19	264.39	3.80	A_2, B_1, C_1	10.37	
0.6	0.4	391.15	387.97	3.17	A_1, B_1, C_1, D_1	4.95	
0.5	0.4	474.74	472.11	2.63	A_3, E_1	7.27	
0.4	0.4	554.07	551.96	2.11	A_2, B_1, C_1	7.29	
0.3	0.4	618.55	616.92	1.63	A_2, B_1, C_1	10.34	
0.2	0.4	774.73	773.62	1.11	A_2, C_1, D_1	5.15	*
0.1	0.4	870.88	870.32	0.55	A_2, C_1, D_1	7.31	*

Table 3 continued

α	γ	*Sol*	*TC* $\times\ \alpha$	*Los* $\times$ $(1-\alpha)$	Vehicles	Time (s)	DSCV[a]
(*b*) *Instance a9–24*							
0.9	0.1	187.14	177.69	9.44	A_4, B_2, C_1, E_1	7.13	
0.8	0.1	363.78	355.38	8.39	A_4, B_2, C_1, E_1	8.82	
0.7	0.1	540.42	533.08	7.34	A_4, B_2, C_1, E_1	5.91	
0.6	0.1	669.57	663.25	6.32	A_4, B_2, C_1, E_1	7.17	
0.5	0.1	893.71	888.46	5.25	A_4, B_2, C_1, E_1	12.57	
0.4	0.1	993.42	989.23	4.19	A_4, B_2, C_1, E_1	10.51	*
0.3	0.1	1247.85	1244.69	3.16	A_4, B_2, C_1, E_1	7.52	
0.2	0.1	1344.61	1342.50	2.11	A_4, B_2, C_1, E_1	8.07	
0.1	0.1	1600.28	1599.23	1.05	A_4, B_2, C_1, E_1	5.51	
0.9	0.2	173.74	164.11	9.63	A_4, B_2, C_1, E_1	3.89	*
0.8	0.2	356.84	348.22	8.61	A_4, B_2, C_1, E_1	7.70	*
0.7	0.2	508.87	501.33	7.54	A_4, B_1, C_1, D_1, E_1	7.07	*
0.6	0.2	658.11	652.00	6.11	A_4, B_2, C_1, E_1	5.23	*
0.5	0.2	830.10	825.00	5.09	A_4, B_2, C_1, E_1	3.34	*
0.4	0.2	1048.94	1044.66	4.28	A_4, B_2, C_1, E_1	4.21	*
0.3	0.2	1144.06	1141.00	3.06	A_4, B_2, C_1, E_1	6.16	*
0.2	0.2	1306.04	1304.00	2.04	A_4, B_2, C_1, E_1	4.51	*
0.1	0.2	1460.07	1458.99	1.08	A_4, B_2, C_1, E_1	7.59	*
0.9	0.3	171.80	162.11	9.69	A_4, B_1, C_1, D_1, E_1	9.99	*
0.8	0.3	334.15	326.00	8.15	A_4, B_2, C_1, E_1	7.55	*
0.7	0.3	493.87	486.33	7.54	A_4, B_1, C_1, D_1, E_1	8.66	*
0.6	0.3	678.11	672.00	6.11	A_4, B_2, C_1, E_1	7.92	*
0.5	0.3	880.10	875.00	5.10	A_4, B_2, C_1, E_1	5.59	*
0.4	0.3	1006.97	1002.66	4.31	A_4, B_1, C_1, D_1, E_1	6.37	*
0.3	0.3	1221.98	1218.77	3.21	A_4, B_2, C_1, E_1	6.88	
0.2	0.3	1315.02	1312.88	2.14	A_4, B_2, C_1, E_1	4.59	
0.1	0.3	1460.07	1458.99	1.08	A_4, B_2, C_1, E_1	6.92	
0.9	0.4	184.57	175.41	9.16	A_4, B_2, C_1, E_1	9.88	*
0.8	0.4	333.41	325.18	8.23	A_4, B_2, C_1, E_1	5.66	*
0.7	0.4	497.35	490.23	7.12	A_4, B_2, C_1, E_1	10.37	*
0.6	0.4	676.54	670.36	6.17	A_4, B_1, C_1, D_1, E_1	4.95	*
0.5	0.4	832.14	827.05	5.09	A_4, B_2, C_1, E_1	7.27	*
0.4	0.4	1056.54	1052.47	4.07	A_4, B_2, C_1, E_1	7.29	*
0.3	0.4	1230.93	1227.88	3.05	A_4, B_2, C_1, E_1	10.34	*
0.2	0.4	1398.79	1396.73	2.06	A_4, B_2, C_1, E_1	5.15	*
0.1	0.4	1579.72	1578.70	1.02	A_4, B_2, C_1, E_1	7.31	*

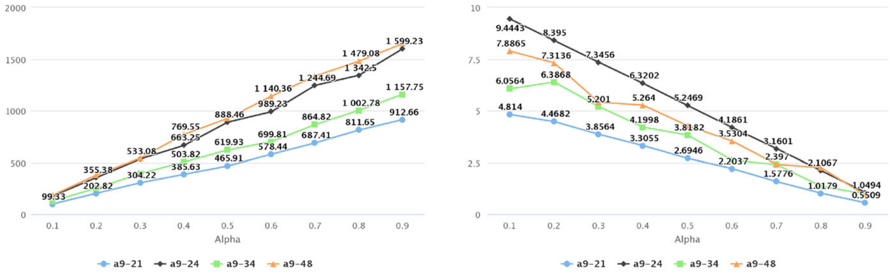
Table 3 continued

α	γ	*Sol*	$TC \times \alpha$	$Los \times (1-\alpha)$	Vehicles	Time (s)	DSCV[a]
(c) *Instance* a9-34							
0.9	0.1	134.59	128.53	6.06	A_2, C_1, D_1, E_1	60.91	
0.8	0.1	258.87	252.48	6.39	A_2, B_1, C_1, D_1, E_1	123.79	
0.7	0.1	398.81	393.61	5.20	A_2, C_1, D_1, E_1	69.98	
0.6	0.1	508.02	503.82	4.20	A_2, C_1, D_1, E_1	93.09	
0.5	0.1	623.75	619.93	3.82	A_3, D_1, E_1	58.69	
0.4	0.1	702.42	699.81	2.61	A_2, D_1, E_1	60.28	
0.3	0.1	867.22	864.82	2.40	A_3, D_1, E_1	54.47	
0.2	0.1	1004.14	1002.78	1.36	A_1, B_1, D_1, E_1	101.48	
0.1	0.1	1158.72	1157.75	0.97	A_2, B_1, D_1, E_1	28.29	
0.9	0.2	135.86	128.86	6.99	A_2, C_1, D_1, E_1	27.72	
0.8	0.2	228.28	223.06	5.23	A_2, D_1, E_1	45.41	*
0.7	0.2	337.61	332.84	4.77	A_2, D_1, E_1	38.17	*
0.6	0.2	447.87	443.78	4.09	A_2, D_1, E_1	35.21	*
0.5	0.2	636.73	632.65	4.07	A_2, C_1, D_1, E_1	21.90	
0.4	0.2	681.34	678.61	2.73	A_1, B_1, D_1, E_1	39.65	*
0.3	0.2	905.16	902.81	2.35	A_2, C_1, D_1, E_1	36.28	
0.2	0.2	954.02	952.72	1.31	A_1, B_1, D_1, E_1	28.16	*
0.1	0.2	1130.77	1129.96	0.81	A_2, C_1, D_1, E_1	36.30	*
0.9	0.3	113.48	107.57	5.91	A_2, D_1, E_1	48.00	*
0.8	0.3	217.15	211.84	5.31	A_2, D_1, E_1	24.01	*
0.7	0.3	335.21	330.52	4.69	A_2, D_1, E_1	18.55	*
0.6	0.3	444.71	440.69	4.02	A_2, D_1, E_1	25.12	*
0.5	0.3	604.76	601.51	3.24	A_2, D_1, E_1	12.43	*
0.4	0.3	656.13	653.45	2.68	A_2, D_1, E_1	29.66	
0.3	0.3	771.58	769.57	2.01	A_2, D_1, E_1	19.97	*
0.2	0.3	861.83	860.52	1.31	A_2, D_1, E_1	25.99	*
0.1	0.3	999.11	998.43	0.68	A_2, D_1, E_1	20.18	
0.9	0.4	103.70	98.49	5.21	A_2, B_1, D_1	12.14	*
0.8	0.4	203.33	197.88	5.45	A_2, D_1, E_1	29.49	*
0.7	0.4	286.59	281.83	4.76	A_3, E_1	18.63	*
0.6	0.4	385.91	381.55	4.36	A_2, B_1, E_1	20.47	*
0.5	0.4	494.83	491.76	3.07	A_2, B_1, D_1	9.64	*
0.4	0.4	593.12	590.75	2.36	A_2, B_1, D_1	12.60	*
0.3	0.4	674.02	671.79	2.23	A_3, B_1	16.21	*
0.2	0.4	738.43	736.96	1.46	A_3, B_1	18.58	*
0.1	0.4	871.85	871.13	0.72	A_2, B_1, D_1	11.02	*

Table 3 continued

α	γ	*Sol*	*TC* $\times$ α	*Los* $\times$ $(1-\alpha)$	Vehicles	Time (s)	DSCV[a]
(*d*) *Instance* a9-48							
0.9	0.1	193.54	185.65	7.89	A_4, B_1, D_1, E_1	307.86	*
0.8	0.1	386.99	379.67	7.31	A_3, B_1, C_1, D_1, E_1	169.86	*
0.7	0.1	546.84	541.43	5.41	A_2, B_2, D_1, E_1	132.84	*
0.6	0.1	774.81	769.55	5.26	A_4, B_1, D_1, E_1	142.75	*
0.5	0.1	929.56	925.29	4.27	A_4, B_1, D_1, E_1	236.38	*
0.4	0.1	1143.89	1140.36	3.53	A_4, C_1, D_1, E_1	81.47	*
0.3	0.1	1337.35	1334.94	2.41	A_4, D_1, E_1	125.41	*
0.2	0.1	1481.33	1479.08	2.25	A_4, C_1, D_1, E_1	133.25	
0.1	0.1	1649.82	1648.90	0.92	A_4, C_1, D_1, E_1	155.19	*
0.9	0.2	175.10	168.57	6.53	A_3, C_1, D_1, E_1	165.68	*
0.8	0.2	348.03	341.77	6.26	A_3, C_1, D_1, E_1	93.12	*
0.7	0.2	500.93	495.56	5.37	A_3, C_1, D_1, E_1	164.11	*
0.6	0.2	661.91	657.25	4.66	A_3, C_1, D_1, E_1	70.64	*
0.5	0.2	846.20	842.50	3.70	A_3, C_1, D_1, E_1	61.88	*
0.4	0.2	999.97	996.92	3.05	A_2, B_1, C_1, D_1, E_1	96.12	*
0.3	0.2	1151.72	1149.44	2.28	A_2, B_1, C_1, D_1, E_1	51.44	*
0.2	0.2	2827.00	1348.00	1.48	A_3, C_1, D_1, E_1	107.39	*
0.1	0.2	1505.42	1504.65	0.77	A_2, B_1, C_1, D_1, E_1	95.47	*
0.9	0.3	154.74	147.93	6.81	A_2, C_1, D_1, E_1	115.34	*
0.8	0.3	288.53	283.51	5.02	A_2, C_1, D_1, E_1	62.70	*
0.7	0.3	452.79	447.61	5.18	A_3, C_1, D_1, E_1	63.38	*
0.6	0.3	570.94	567.15	3.79	A_2, C_1, D_1, E_1	39.10	*
0.5	0.3	737.31	733.99	3.32	A_2, C_1, D_1, E_1	127.30	*
0.4	0.3	863.69	861.16	2.53	A_1, B_1, C_1, D_1, E_1	65.40	*
0.3	0.3	994.52	992.62	1.90	A_2, C_1, D_1, E_1	64.00	*
0.2	0.3	1159.17	1157.94	1.23	A_2, B_1, C_1, D_1	39.37	*
0.1	0.3	1343.53	1342.90	0.63	A_2, C_1, D_1, E_1	39.64	*
0.9	0.4	147.10	141.56	5.54	A_2, C_1, D_1, E_1	67.86	*
0.8	0.4	284.40	279.45	4.95	A_2, B_1, C_1, D_1	36.36	*
0.7	0.4	421.06	414.76	6.30	A_4, D_1	17.28	*
0.6	0.4	565.69	561.32	4.37	A_3, D_1, E_1	33.47	*
0.5	0.4	710.43	707.36	3.07	A_2, C_1, D_1, E_1	31.24	*
0.4	0.4	846.01	843.51	2.50	A_4, D_1	31.53	*
0.3	0.4	987.06	984.71	2.35	A_2, B_1, C_1, D_1	28.90	*
0.2	0.4	1076.28	1075.04	1.24	A_3, B_1, D_1	37.31	*
0.1	0.4	1269.22	1268.57	0.65	A_3, D_1, E_1	18.00	*

[a] Dynamic solution with constraint violation



(a) Transportation Costs **(b)** User Inconvenience

Fig. 4 Approximation to the pareto frontier of transportation costs and user inconvenience with 10% of dynamism for all instances

We also note that the worst of the violations occurs in the instance a9-48 (Table 3d). In this instance, the total waiting for patients is equal to 6.80 minutes, and the sum of delays concerning the time window is 51.02 minutes. As can be seen, this time may not be significant to justify the rejection of the requests, as in the HHDARP model.

From the results, we can see that as α decreased, the transportation cost grew, and the user inconvenience declined in all instances. To better visualize the trade-off between the two conflicting objectives, transportation cost and user inconvenience, we plotted (TC) and *(Los)* for all instances and $\gamma = 0.1$ (see Fig. 4). We can also observe that the objective function value grew when α decreased, whereas the objective function value lowered when γ increased. This occurs because α is the weight associated with TC and TC is the parcel that contributes most to the objective function. On the other hand, an increment in γ means that the instance has fewer static requests to be attended in the first stage, lowering the *sol* value. As expected, increasing γ also caused a higher number of time window violations. It is more difficult to respect the time window constraints when there are more dynamic requests. We can notice that the number of solutions with violations was larger in instances a9-24, a9-34, and a9-48, with γ values 0.3 and 0.4.

6.3 Comparison between Scenario 1 and Scenario 2

Comparing the results obtained in Scenario 1 and 2, we noticed that although the number of a-priori known (static) transportation requests in Scenario 2 is smaller than that in Scenario 1, the number of vehicles used in the route plan is almost the same. This occurs due to the different nature of services and different vehicle types (see Fig. 1). Therefore, in most of the solutions more than one type of vehicle was used in a route plan (see column *Vehicle* in Table 2 and Tables 3a–d). We also conclude that by including the dynamism in the designed instances, solutions presented violations. In Scenario 2, some route plans had time window violations compared to no solution with time window violations in Scenario 1. Therefore, the presence of urgent transportation requests caused violations in the solutions. In this way, we can infer that the higher the dynamism is, the higher the probability of a violation occurring in a solution is.

7 Conclusions

In this paper, we proposed the bi-objective heterogeneous dynamic dial-a-ride problem with no rejection for the real problem faced by a patient transportation request system in Brazil. In this problem, all transportation requests must be attended and there are two conflicting goals: the minimization of the transportation costs, which consists of routing and vehicle costs, and the minimization of user inconvenience. Therefore, time window constraints are relaxed, and if a user cannot be delivered within the time window, we penalize the objective function value by increasing user inconvenience. To validate our model, we built a two-phase heuristic algorithm. The first phase uses a GVNS algorithm with RVND as the local search procedure. This phase solves the static part of the problem. To solve the dynamic part of the problem, we use a simple insertion heuristic. For analyzing the effect of dynamism in the problem, we create two scenarios, one with no dynamic requests and the other with a certain amount of dynamic requests (Scenarios 1 and 2). From our experiments, we concluded that by including the dynamism, i.e., increasing the number of dynamic requests, the number of vehicles adopted in a route plan increases. Attending only the static customers in Scenario 2 required the same number of vehicles than attending all the customers in Scenario 1 for most of the instances. We can also infer that an increment in the number of dynamic requests causes a higher probability of time window violation occurrences. The solution provided by the heuristic algorithm requires fewer vehicles than the route plan currently adopted by the transportation sector of the Ouro Preto' health care system. The algorithm also calculated route plans with lower traveled time and user waiting time. Moreover, we built an approximation to the Pareto frontier for each instance. In this way, the transportation sector can select the best route plan based on their preferences (e.g., favoring the minimization of user inconvenience).

Acknowledgements This work was partly done when the first author was a visiting researcher at the University of Bremen thanks to the Bremen IDEA scholarship. The authors would like to thank the National Council for Scientific and Technological Development (CNPq) for a Doctoral Degree scholarship granted to the second author and also for the support to the third and last authors (Grants 438473/2018-3 and 303266/2019-8, respectively). We also thank the Minas Gerais State Foundation for Research Support (FAPEMIG, grant CEX-PPM-00676-17), the Federal University of Ouro Preto (UFOP), and the Coordination for the Improvement of Higher Education Personnel (CAPES)—Finance Code 001—for supporting the development of the present study. We want to express special thanks to the transportation sector of the Ouro Preto' health care system for providing information for modeling the problem addressed.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Beaudry, A., Laporte, G., Melo, T., Nickel, S.: Dynamic transportation of patients in hospitals. OR Spectrum **32**(1), 77–107 (2010)
2. Bernardo, M., Pannek, J.: Robust solution approach for the dynamic and stochastic vehicle routing problem. J. Adv. Transp. 1–11 (2018)

3. Campbell, A.M., Savelsbergh, M.: Efficient insertion heuristics for vehicle routing and scheduling problems. Transp. Sci. **38**(3), 369–378 (2004)
4. Cordeau, J.F.: A branch-and-cut algorithm for the dial-a-ride problem. Oper. Res. **54**(3), 573–586 (2006)
5. de Freitas, J.C., Penna, P.H.V.: A variable neighborhood search for flying sidekick traveling salesman problem. Int. Trans. Oper. Res. **27**(1), 267–290 (2020). https://doi.org/10.1111/itor.12671
6. Hanne, T., Melo, T., Nickel, S.: Bringing robustness to patient through optimized patient transports in hospitals. Interfaces **39**(3), 241–255 (2009)
7. Hansen, P., Mladenović, N., Brimberg, J., Pérez, J.A.: Variable neighborhood search. Int. Ser. Oper. Res. Manag. Sci. **24**, 1097–1100 (1997)
8. Hansen, P., Mladenović, N., Todosijević, R., Hanafi, S.: Variable neighborhood search: basics and variants. EURO J. Comput. Optim. **5**(3), 423–454 (2017). https://doi.org/10.1007/s13675-016-0075-x
9. Ho, S.C., Szeto, W.Y., Kuo, Y.H., Leung, J.M.Y., Petering, M., Tou, T.W.H.: A survey of dial-a-ride problems: literature review and recent developments. Transp. Res. Part B Methodol. **111**, 395–421 (2018)
10. Karakostas, P., Sifaleras, A., Georgiadis, M.C.: A general variable neighborhood search-based solution approach for the location-inventory-routing problem with distribution outsourcing. Comput. Chem. Eng. **126**, 263–279 (2019). https://doi.org/10.1016/j.compchemeng.2019.04.015
11. Karakostas, P., Sifaleras, A., Georgiadis, M.C.: Adaptive GVNS heuristics for solving the pollution location inventory routing problem. Lecture Notes in Computer Science pp. 157–170 (2020). https://doi.org/10.1007/978-3-030-38629-0
12. Madsen, O.B.G., Ravn, H.F., Rygaard, J.M.: A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. Ann. Oper. Res. **60**(1), 193–208 (1995)
13. Penna, P.H.V., Subramanian, A., Ochi, L.S., Vidal, T., Prins, C.: A hybrid heuristic for a broad class of vehicle routing problems with heterogeneous fleet. Ann. Oper. Res. **273**(1), 5–74 (2019). https://doi.org/10.1007/s10479-017-2642-9
14. Schilde, M., Doerner, K.F., Hartl, R.F.: Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. Comput. Oper. Res. **38**(12), 1719–1730 (2011)
15. Souza, M.J.F., Coelho, I.M., Ribas, S., Santos, H.G., Merschmann, L.H.C.: A hybrid heuristic algorithm for the open-pit-mining operational planning problem. Eur. J. Oper. Res. **207**(2), 1041–1051 (2010)
16. Subramanian, A., Drummond, L.M.A., Bentes, C., Ochi, L.S., Farias, R.: A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. Comput. Oper. Res. **37**(11), 1899–1911 (2010)

Affiliations

André L. S. Souza[1] · Marcella Bernardo[2] · Puca H. V. Penna[1] · Jürgen Pannek[3] · Marcone J. F. Souza[1]

✉ André L. S. Souza
andre.souza@aluno.ufop.edu.br

Marcella Bernardo
marcella@uow.edu.au

Puca H. V. Penna
puca@ufop.edu.br

Jürgen Pannek
pan@biba.uni-bremen.des

Marcone J. F. Souza
marcone@ufop.edu.br

[1] Departamento de Computação, Universidade Federal de Ouro Preto, Ouro Preto 35400-000, Brazil

[2] SMART Infrastructure Facility, University of Wollongong, Northfields Avenue, Wollongong 2522, Australia

[3] BIBA Bremer Insitut für Produktion und Logistik GmbH, University of Bremen, Hochschulring 20, 28359 Bremen, Germany