INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 0 (2024) 1–30
DOI: 10.1111/itor.13585

Multi-neighborhood simulated annealing for the home healthcare routing and scheduling problem

Sara Ceschia, Luca Di Gaspero, Roberto Maria Rosati and Andrea Schaerf*

DPIA, Università degli Studi di Udine, Via delle Scienze 206, Udine I-33100, Italy
E-mail: sara.ceschia@uniud.it[Ceschia]; luca.digaspero@uniud.it[Di Gaspero]; robertomaria.rosati@uniud.it[Rosati]; andrea.schaerf@uniud.it[Schaerf]

Received 5 April 2024; received in revised form 1 October 2024; accepted 19 November 2024

Abstract

Over time, the focus on supportive and geriatric care has shifted from being predominantly provided in institutional settings like nursing or rest homes to be delivered within the homes of the patients. Trained caregivers now provide home healthcare services by visiting patients in their own homes and carrying out specific services based on each patient's individual needs before moving on to the next patient. Planning such a service involves considering the routing aspect and ensuring synchronization between services and designated time windows for patients. To solve the problem, we propose a local search approach that combines different neighborhood operators guided by the simulated annealing metaheuristic. Additionally, we introduce a realistic and diverse dataset and a robust and flexible file format based on JSON. This dataset and format have the potential to facilitate future comparisons and analyses. Our study shows that by appropriately tuning our algorithm in a statistically rigorous manner, it outperforms existing methods on all benchmarks.

Keywords: home healthcare; routing and scheduling with time windows; route synchronization; neighborhood search; simulated annealing

1. Introduction

In recent times, we have witnessed a shift in the provision of supportive and geriatric care. Instead of being primarily based in institutional settings like nursing or rest homes, these services have been increasingly relocated to the patient's own homes.

* Corresponding author.

The reasons behind this shift can be attributed to two factors. First, allowing patients to remain in their familiar environments enhances their quality of life. Second, this situation has a notable positive effect on reducing healthcare costs (Genet et al., 2012).

Trained caregivers provide home healthcare services by visiting the patient's home during a designated time. They carry out service operations based on the patient's specific needs, ranging from medical care to instrumental activities of daily living. After completing their tasks, the caregivers proceed to the next patient. This distinct characteristic of home healthcare makes it a structured problem. Unlike classical activity scheduling problems in hospitals or healthcare institutions, which primarily focus on temporal aspects, home healthcare requires considering both temporal and spatial dimensions, including travel times between patients.

The synchronization constraint stands out as the most distinctive type of constraint in this problem, as it introduces temporal dependencies among the activities. For instance, specific medical care tasks such as physiotherapy necessitate the simultaneous presence of multiple caregivers, such as when lifting the patient from their bed. Other activities, such as administering medication or preparing lunch, may require subsequent actions to take place after a specific time interval, like administering one dose in the morning and another in the evening. Due to the nonnegligible travel time involved when caregivers transition from one patient to another, these constraints significantly increase the complexity of the routing aspect of the problem compared to standard vehicle routing problems.

This study presents a novel metaheuristic technique to solve the problem formulation introduced by Mankowska et al. (2014). Our approach employs local search and incorporates a diverse set of original neighborhood operators. The core of the solution process relies on implementing a simulated annealing (SA) procedure (Kirkpatrick et al., 1983) for driving the search.

The approach outperforms previous methods in the literature (Mankowska et al., 2014; Lasfargeas et al., 2019; Kummer et al., 2020; Kummer, 2021; Kummer et al., 2024) and shows its capability to improve the current state-of-the-art results on the benchmark instances.

In addition, we have designed a parametric instance generator that relies on real geographic data and real population density. This generator allows us to create an additional dataset that encompasses a significantly wider range of sizes and feature' values compared to the existing benchmarks proposed by Mankowska et al. (2014) and Kummer (2021). For our new dataset, we have introduced a novel file format based on JSON, which offers enhanced robustness, extensibility, and human readability when compared to the previous formats utilized, thus far.

We also show our results on the new instances, highlighting the relationship between the different objectives.

Our public repository at https://github.com/iolab-uniud/hhcrsp contains all validation instances in the new format, our best solutions, and the validator. These resources are provided for examination and future comparisons. The source code and instance generator will also be available in the same repository.

The remainder of the paper is structured as follows: Section 2 provides an overview of related work concerning home healthcare services. Section 3 presents a detailed explanation of the problem formulation. The metaheuristic solution approach is outlined in Section 4, illustrating how the problem is modeled within the local search framework. In Section 5, we describe the available instances and our generator and introduce new instances. The experimental analysis of the proposed solution method, along with the tuning process details, is documented in Section 6, where a comparison with alternative approaches is presented. Lastly, Section 7 offers some concluding remarks and outlines potential avenues for future work.

A preliminary and abridged version of this work appeared in Ceschia et al. (2023), which considers only one neighborhood operator, includes experiments only on the first dataset and does not contain the new data format, the generator, and the validator.

2. Related work

Research on scheduling home healthcare services has been ongoing since the late 1990s. Notably, Begur et al. (1997) were among the first to consider this problem and applied a simple scheduling heuristic as its solution method. Subsequently, Cheng and Rich (1998) formulated the problem using mixed-integer linear programming techniques.

Later attempts to address this problem approached it from a *set covering* perspective. For instance, Eveborn et al. (2006) introduced a working healthcare planning system, while Rasmussen et al. (2012) were pioneers in considering temporal dependencies among the activities. Bredström and Rönnqvist (2008) also worked on a similar synchronization problem but looked at it from a vehicle routing perspective without explicitly considering the healthcare nature of the services involved.

The idea of providing services in a metropolitan area, where the travel between patients can be facilitated using a public transportation network instead of a dedicated car, has been explored in the studies conducted by Bertels and Fahle (2006) and Rendl et al. (2012). The latter research also considers switching the mode of transportation during the journey, resulting in multimodal trips.

Di Gaspero and Urli (2014) tackled a similar problem, albeit with a distinct perspective. Their study focused on a temporal horizon that extended over multiple days. They also incorporated the balancing of caregivers' workloads, aiming to minimize overtime. Additionally, their approach allowed for the possibility of leaving certain patient activities unscheduled, which proved helpful in handling overconstrained scenarios and allowed for hiring more occasional caregivers if needed. To address this problem, they modeled it in the constraint programming framework and employed specialized branching heuristics and a large neighborhood search approach to solve it.

Mankowska et al. (2014) presented a relatively standardized representation of the problem and offered a set of benchmark instances for evaluation purposes. The problem formulation (refer to Section 3 for an informal overview) incorporates synchronization constraints at patients' homes, allowing for a maximum of two activities to be coordinated. The quality of the solution is assessed based on travel times and the tardiness of service activities concerning patients' time windows. To solve this problem, Mankowska et al. (2014) devised an adaptive variable neighborhood search method incorporating eight distinct neighborhood operators.

A similar problem was addressed by Ait Haddadene et al. (2016), but in their formulation, patients' time windows must be strictly respected, and the objective is to minimize total traveling times and patients' nonpreferences related to caregivers. A Mixed Integer Programming (MIP) model and a hybrid greedy randomized adaptive search procedure and iterated local search metaheuristics are proposed and compared on the testbed designed by Bredström and Rönnqvist (2008), conveniently extended to take into account different types of services. Several metaheuristics for this formulation were also proposed by Masmoudi et al. (2023).

Recent advancements in the field include the research conducted by Lasfargeas et al. (2019). They proposed a local search-based method integrated into a variable neighborhood search solution procedure for a multiperiod case and evaluated it also on benchmarks by Mankowska et al. (2014). Various researchers have also explored population-based approaches tailored to different problem settings. For instance, Decerle et al. (2018) and Grenouilleau et al. (2019) developed memetic algorithms, which involve genetic algorithms followed by a local search step, to address the problem. On the other hand, Clapper et al. (2023) introduced a model-based evolutionary algorithm. Following a similar approach to the work by Di Gaspero and Urli (2014), Grenouilleau et al. (2019) considered a multiday horizon and aimed to balance caregivers' workload, including minimizing overtime. However, their routing problem was more complex, accounting for hourly dependent traveling times due to traffic. Moreover, Xiang et al. (2021) and Oladzad-Abbasabady et al. (2023) approached the problem from a multiobjective perspective, seeking to balance the total operating cost and the satisfaction of caregivers and patients. They employed NSGA-II genetic and iterated local search algorithms to tackle the problem. Kordi et al. (2023) considered four objectives, namely total cost, environmental emission, workload balance, and service quality and proposed a multiobjective variable neighborhood search approach.

The problem formulation introduced by Mankowska et al. (2014) has attracted interest from Kummer and co-workers (Kummer et al., 2020; Kummer, 2021; Kummer et al., 2024), who developed and utilized biased random key genetic algorithms to address the problem. Their outcomes currently stand at the state-of-the-art level concerning these benchmarks. Additionally, in their work, Kummer (2021) introduced a new dataset with a more comprehensive set of features than the original dataset proposed by Mankowska et al. (2014).

Rich formulations that include real-world constraints were presented by Liu et al. (2021) and Bazirha et al. (2023a). Liu et al. (2021) consider lunch breaks for caregivers, flexible departure locations, and synchronized visits. The formulation of Bazirha et al. (2023a) considers patients with multiple time windows requiring one or more services that must be delivered simultaneously or independently. In both works, the problems are first modeled by mixed integer programming and then solved using metaheuristic methods.

Recently, the home healthcare routing and scheduling problem with stochastic travel and service times was examined by Bazirha et al. (2023b). They formulated the problem as a two-stage stochastic programming model with recourse, which involves penalty costs for delayed services to patients and remuneration for caregivers' extra working time. The researchers employed CPLEX, a genetic algorithm, and variable neighborhood search-based heuristics to solve the deterministic model. For the stochastic programming model, they utilized Monte Carlo simulation embedded into the genetic algorithm. The performance of these solution methods was evaluated on test instances generated following the benchmark instances proposed by Mankowska et al. (2014).

Notable surveys on the utilization of operational research methods in the context of home, healthcare problems are presented in the works of Fikar and Hirsch (2017), Cissé et al. (2017), Grieco et al. (2021), and Di Mascolo et al. (2021).

As noted by Soares et al. (2024, Section 4.2) in their recent survey on synchronization in vehicle routing, Home Healthcare is arguably the most relevant application of routing with synchronization schedules, and, in comparison to the classic routing problems, it has its peculiar predominant aspects. The most relevant are (see also Di Mascolo et al., 2021): (i) capacity constraints are not considered; (ii) due to the high degree of specialization of home healthcare tasks, not all caregivers

Table 1
Abbreviations of objectives and constraints for home healthcare routing and scheduling problems.

Objective abbreviation	Description	Constraint abbreviation	Description
FA	Fairness (work balance)	CB	Caregiver break
OT	Overtime	MC	Multiple centers
TC	Travel cost	PR	Preference
TT	Travel time	PC	Precedence
UP	Unvisited patients	SY	Synchronization
WT	Waiting time	SK	Skill requirements
		TW	Time windows
		UT	Uncertain travel and service times
		WR	Working time regulations

Table 2
Objectives and constraints of single-period home healthcare routing and scheduling problems

	Objectives						Constraints							
Reference	FA	OT	TC	TT	UP	WT	CB	MC	PR	PC	SY	SK	TW	UT
Mankowska et al. (2014)	✓	✓		✓						✓	✓	✓	✓	
Hiermann et al. (2015)		✓		✓		✓		✓	(✓)			(✓)	(✓)	
Ait Haddadene et al. (2016)				✓					(✓)	✓	✓	✓	✓	
Liu et al. (2017)			✓		✓		✓		✓				✓	
Decerle et al. (2018)				✓				✓			(✓)	✓	(✓)	
Parragh and Doerner (2018)			✓		✓			✓			✓	✓	✓	
Liu et al. (2019)			✓		✓			✓				✓	(✓)	✓
Liu et al. (2021)			✓				✓	✓			✓	✓	✓	
Xiang et al. (2021)			✓						(✓)			✓	✓	
Bazirha et al. (2023b)		✓	✓								✓	✓	✓	✓
Bazirha et al. (2023a)	✓					✓					✓	✓	✓	
Clapper et al. (2023)		✓	✓			✓			(✓)			✓	✓	
Kordi et al. (2023)	✓		✓						(✓)			✓	✓	
Oladzad-Abbasabady et al. (2023)		✓	✓		✓	✓		✓	(✓)	(✓)	(✓)	✓	(✓)	

may be able to perform certain services; (iii) services to the same patient should be performed by the same caregivers (continuity of care); and (iv) there are fairness-related issues, such as balanced workloads. Other examples of routing with synchronization schedules can be found in vessel scheduling at ports (Bakkehaug et al., 2016), gantry crane operations in railway transport (Fedtke and Boysen, 2017), and truck routing and scheduling in public works (Grimault et al., 2017).

We conclude this section by providing in Table 2 an overview of objectives and constraints of single-period home healthcare routing and scheduling formulations recently proposed in the literature (for previous references see Fikar and Hirsch, 2017). Table 1 lists the features considered and their abbreviations used in Table 2. The ✓ symbol means that the specific feature is considered in the formulation proposed in the article. It is in parentheses when the constraint is not considered hard and is included in the objective function as a soft constraint.

3. Problem definition

The specific formulation of the home healthcare routing and scheduling problem considered in this work has been introduced and formally defined by using a mathematical model by Mankowska et al. (2014, Section 3). For completeness, we present a summary of its definition in this paper.

The primary entities involved in the problem are as follows:

Times and horizon: The problem consists of planning for a single day, with time expressed in minutes, starting from 0, which corresponds to the beginning of daily activities (e.g., 6:00 a.m.) when all caregivers are assumed to be at the *central office*. There is no explicit time horizon, and the patient's time windows limit the activities. Distances between patients and from the central office are directly measured in minutes required to travel and cover those distances.

Patients: Patients are categorized into two classes: *single-service* patients and double-service ones. Single-service patients require service from one caregiver within their designated time window. In contrast, double-service patients need to be served by two caregivers, either *simultaneously* or *sequentially*, within their time window. For sequential double-service patients, the minimum and maximum time gap between the two services is explicitly specified.

Services and caregivers: Every service has its duration, which may differ based on the individual patient's requirements. Each caregiver specializes in providing a particular service and leveraging their specific abilities. The caregivers start their daily tasks from the central office and complete their workday by returning to the exact central location.

According to the model of Mankowska et al. (2014), service time can vary depending on the specific service and the patient, even though this length is a global constant value in the original dataset. In the dataset of Kummer (2021), the values differ between services and patients, although they do not depend on the specific caregiver delivering the service. In constructing our dataset, we also assume that the services provided to patients rely on the type of service and the patient rather than the caregiver themselves.

Figure 1 shows the data of a toy instance with six patients (p_1, p_2, …, p_6), three caregivers (c_1, c_2, and c_3), and three services (s_1, s_2, and s_3). As mentioned above, distances, service times, and time windows are expressed in minutes. For example, assuming the working day starts at 6:00 a.m., the first time window 240–360 represents that patient p_1 should be visited between 10:00 a.m. and 12:00 a.m.. Furthermore, p_1 needs a single service s_2, whose duration is 30 minutes. Conversely, patients p_4 and p_5 require a double service: specifically, p_4 needs the simultaneous presence of two caregivers, whereas for p_5 the two services must be separated by at least 30 minutes and, at most, 45 minutes. The central office is denoted as *co* in the distance matrix. Notice that distances are not symmetric.

The possibility that a single caregiver provides both services for the same double-service patient is impossible for simultaneous services. It is also explicitly forbidden for sequential ones in the

Patients

ID	Time window	Service (duration) first	second	Separation min	max
p_1	240 – 360	s_2 (30)	—	—	
p_2	120 – 180	s_3 (20)	—	—	
p_3	0 – 60	s_2 (45)	—	—	
p_4	120 – 210	s_2 (30)	s_3 (30)	0	0
p_5	270 – 420	s_1 (15)	s_3 (30)	30	45
p_6	360 – 420	s_1 (45)	s_3 (20)	60	90

Caregivers

ID	Services
c_1	s_1, s_2
c_2	s_3
c_3	s_2, s_3

Distances

co	0	38	34	56	7	13	26
p_1	39	0	23	22	32	50	58
p_2	35	23	0	44	28	47	43
p_3	56	22	44	0	54	59	78
p_4	7	32	28	51	0	19	28
p_5	13	45	47	59	19	0	35
p_6	27	57	42	77	28	35	0
	co	p_1	p_2	p_3	p_4	p_5	p_6

Fig. 1. Toy instance data.

work by Mankowska et al. (2014). From a practical standpoint, it may not be intuitive to impose this limitation, as using a single caregiver has the potential to save time in the overall schedule. However, in real-life scenarios, it is assumed that the two services (e.g., nursing and physiotherapy) require distinct skills and qualifications. Therefore, it is highly improbable that a single caregiver possesses both sets of skills. In the available datasets, the situation that a single caregiver has the ability for two sequential services for the same patient never happens, so this is not an option. We keep the same limitation and this situation does not occur in our dataset either (notice that the toy instance, due to the limited size, does not enforce this limitation).

Enforcing the caregiver's qualification for the specific service is a hard constraint. Likewise, it is strictly prohibited for a service to commence before a patient's designated time window begins. Following standard vehicle routing practices, if the caregiver arrives early, they must wait until the patient's time window starts. On the other hand, a patient being served late is admissible, and the extent of tardiness is considered in the objective function. Finally, the minimum and maximum time separation between sequential services for a double-service patient is also a hard constraint.

Notice that it is also possible that some caregivers are not assigned to any patient; thus, they have an empty route. Conversely, all patients must be covered for all the services they need.

The problem goal is to minimize an objective function that consists of three components: the total travel time, the overall tardiness, and the highest individual tardiness. Including the highest tardiness component is essential to ensure fairness among patients. This approach prevents the

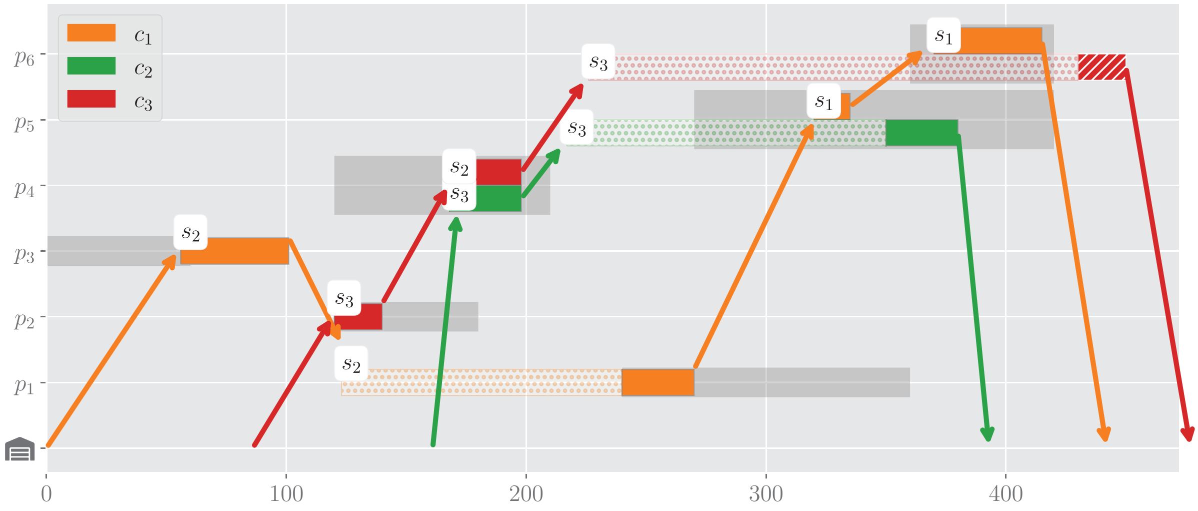

Fig. 2. A solution of the toy instance.

solution from being optimized at the expense of significantly delaying service to a single patient. In cases where a patient receives a double service, each service tardiness is considered independently. All three components are measured in minutes and are combined with equal weight, each contributing $\frac{1}{3}$ to the overall objective.

Figure 2 shows a (nonoptimal) solution to the instance of Fig. 1, in the visual form automatically produced by our solution validator. The time windows of patients are highlighted in dark gray, and the warehouse icon refers to the central office. The travel times of caregivers c_1, c_2, and c_3 are 190 (56 + 22 + 50 + 35 + 27), 39 (7 + 19 + 13), and 117 (34 + 28 + 28 + 27), respectively. Notice that all services are provided on time, except for service s_3 by caregiver c_3 at patient p_6 (shown with white diagonal stripes in the figure) that is late by 10 minutes. Hence, the total and highest lateness values are both 10. Consequently, the cost of this solution is calculated as $(190 + 39 + 117)/3 + 10/3 + 10/3$, resulting in a total cost of 122. Notice also that three early arrivals have long waiting times (shown in dotted rectangles in the figure). This phenomenon also happens in other instances but is particularly evident in this toy instance.

4. Solution method

We present a local search-based approach to address the problem. In the subsequent sections, we will outline the fundamental components of the local search paradigm, which include the *search space*, the *initial solution strategy*, the *neighborhood relations*, and the *metaheuristic* that drives the search process.

In the following, we denote by $\mathcal{P} = \{1, \ldots, P\}$ the set of patients, $\mathcal{C} = \{1, \ldots, C\}$ the set of caregivers, and $\mathcal{S} = \{1, \ldots, S\}$ the set of services.

4.1. Search space

Following Mankowska et al. (2014), the search space is represented by a vector $\Pi = [\pi_1, \pi_2, \ldots, \pi_P]$ containing a permutation of the values $1, \ldots, P$, which represents a global ordering of the patients. That is, patient π_1 is the first patient to be served, π_2 is the second, and so on.

Vector Π is completed by another P-sized vector of pairs $\Theta = [(\vartheta_{1,1}, \vartheta_{1,2}), (\vartheta_{2,1}, \vartheta_{2,2}), \ldots, (\vartheta_{P,1}, \vartheta_{P,2})]$ such that $\vartheta_{p,1}$ and $\vartheta_{p,2}$ are the caregivers that serve patient p. In the case of a single-service patient p, the second element of the pair ($\vartheta_{p,2}$) is not used.

For example, the solution of Fig. 2 to the instance of Fig. 1 is represented by the vectors: $\Pi = [3, 2, 4, 1, 5, 6]$ and $\Theta = [(1, -), (3, -), (1, -), (3,2),(1,2),(1,3)]$, the dash symbol $-$ means the value is not present.

The routes of caregivers and their corresponding service times are deterministically constructed, starting from sets Π and Θ. The *scheduling procedure* starts with empty routes and processes patients one by one, following the order specified in Π. During each iteration i of this procedure, patient $p = \pi_i$ is added to the end of caregivers' routes $c_1 = \vartheta_{p,1}$ and $c_2 = \vartheta_{p,2}$ (or only to c_1 for patients requiring a single service).

The service start times are calculated *at the earliest*, taking into consideration the time window of patient p. Specifically, for caregiver c_1 at patient p, the start time is determined as the maximum value between the beginning of the time window for p and the earliest time at which c_1 can reach p based on their previous assigned duties and travel times. For the second service (if applicable), the service start time of caregiver c_2 at patient p is determined as the maximum value between the minimum separation between the two services for p and the earliest time at which c_2 can reach the patient.

The solution of Fig. 2 is obtained by applying the scheduling procedure. We can see that all services are provided always as soon as possible.

In the case in which c_2 reaches patient p later than the maximum separation between the two services of p, the service of c_1 is postponed as much as necessary to keep the separation within the range.

This case happens for the solution of the toy instance represented by $\Pi = [4, 3, 1, 5, 2, 6]$ and $\Theta = [(3, -), (2, -), (3, -), (1, 2), (1, 3), (1, 2)$, which incidentally is optimal, with no tardiness and a total traveling time of 334 minutes, corresponding to a cost of 111.33. The schedule built for this solution by the above procedure is shown in Fig. 3, where we notice that caregiver c_1 (orange) at patient p_5 is postponed by 5 minutes to time 275 from the beginning of the time window at 270 because caregiver c_3 can arrive at the patient only at time 320, and the maximum separation is 45 minutes. Notice also that there is no tardiness as long as the caregiver starts within the time window, even though the service finishes after the end of the time window (see caregiver c_2 at patient p_6).

This scheduling procedure always satisfies separations, although tardiness is possible. Treating tardiness as a soft constraint ensures that all necessary services can be accommodated within the schedule, guaranteeing the feasibility of the solution.

The two vectors Π and Θ are sufficient to represent the solution and build the full schedule using the abovementioned procedure. Nonetheless, in our implementation, they are complemented by many redundant data structures used to accelerate the evaluation of the costs of neighbor solutions. These data structures include, among others, the position of each patient in the ordering (the inverse of Π), the route of each caregiver, and the positions of the patients in the routes of their caregivers.

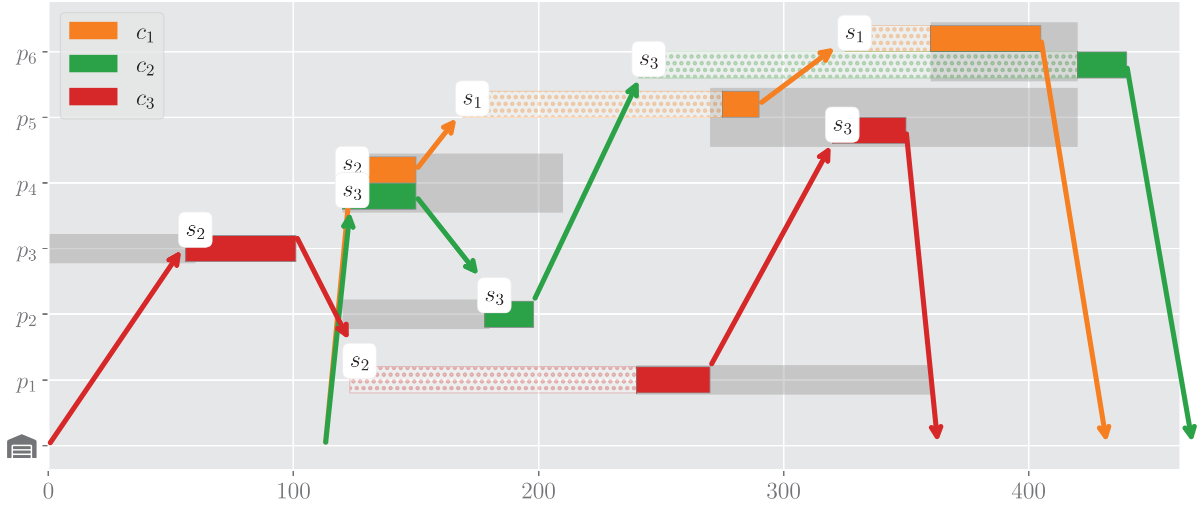

Fig. 3. The optimal solution of the toy instance.

4.2. Initial solution

The initial solution is created randomly. Initially, a random permutation is chosen for the Π vector. Next, one (or two distinct) caregivers are randomly selected for each patient and added to the Θ vector. The caregivers are chosen from those with the required ability to provide the service.

Given the vectors Π and Θ, the scheduling procedure is applied to produce the full initial schedule and evaluate the corresponding costs.

4.3. Neighborhood relations

We consider three neighborhood relations: MovePatient, which repositions and reassigns one patient, SwapPatients, which swaps two patients in terms of global position and caregivers, and InRouteSwap, which swaps two patients within a specific route.

The set of neighborhoods we employ has a significantly broader scope compared to the one utilized by Mankowska et al. (2014). Our neighborhoods enable simultaneous changes in both the position and caregivers, whereas Mankowska et al. (2014) considered only the options of repositioning a patient within the global order, changing the caregiver(s), *or* swapping either the position or caregiver(s) individually.

For the sake of clarity, we illustrate the neighborhoods only for the case of double-service patients, which is the most complex one. The case of single-service ones is obtained simply by ignoring the second caregiver and adjusting the various features accordingly.

Furthermore, throughout this section, to simplify the presentation, given a solution and a patient p, we use the following notation. We call c_1^p and c_2^p the caregivers assigned to p, that is, $c_1^p = \vartheta_{p,1}$, $c_2^p = \vartheta_{p,2}$; we call i^p the current position of p in Π; finally, we call s_1^p and s_2^p the services requested by patient p.

4.3.1. Neighborhood MovePatient

The first neighborhood, called MovePatient (MP), consists of repositioning one patient p in the global ordering *and* assigning new caregivers to p, in one single move:

- *Attributes*: $\langle p, i, c_1, c_2 \rangle$, with $p \in \mathcal{P},\ 1 \le i \le P,\ c_1, c_2 \in \mathcal{C}$
- *Preconditions*:
  - No null moves: $\langle i, c_1, c_2 \rangle \neq \langle i^p, c^p_1, c^p_2 \rangle$
  - Distinct caregivers: $c_1 \neq c_2$
  - No missing abilities: c_1 and c_2 have the ability for the services
- *Effects*: Patient p is moved to position i in the global ordering Π. All patients in positions between i and i_p are shifted accordingly (either forward or backward). Caregivers c_1 and c_2 are assigned to p. The full schedule is recomputed using the defined procedure.
- *Special cases*: The position i can be the same as i_p, which means the move only results in a change of caregivers. Likewise, one or both caregivers can stay unchanged, indicating that the move represents only a modification in the sequence of the routes.

As an example, consider the state in Fig. 3, represented by $\Pi = [4, 3, 1, 5, 2, 6]$ and $\Theta = [(3, -), (2, -), (3, -), (1, 2), (1, 3), (1, 2)]$. Now, take the move MP$\langle 5, 2, 2, 3 \rangle$, which relocates patient 5 to position 2 and assigns caregivers 2 and 3. The application of this move to the given state would lead to the new state $\Pi = [4, \mathbf{5}, 3, 1, 2, 6]$ and $\Theta = [(3, -), (2, -), (3, -), (1, 2), (\mathbf{2}, \mathbf{3}), (1, 2)]$, where the affected values are highlighted in boldface. Notice that the second caregiver (3) is left unchanged.

4.3.2. Neighborhood SwapPatients

The second neighborhood, called SwapPatients (SP), consists of swapping the position in Π and the caregivers in Θ for two patients. A swap is possible only between patients with the same number of services and with current caregivers with the required abilities for the other patient.

For double-service patients, the neighborhood also includes the option that first and second caregivers are crossed between the two patients: the first caregiver of one patient is assigned to the second service of the other patient. This option is stored in a Boolean attribute of the move, called CS (for *cross swap*), such that if CS $=$ F the caregivers are swapped position-wise, whereas if CS $=$ T they are inverted. The definition of the neighborhood is the following:

- *Attributes*: $\langle p_1, p_2, CS \rangle$, with $p_1, p_2 \in \mathcal{P},\ CS \in \{\mathsf{F}, \mathsf{T}\}$
- *Preconditions*:
  - No null moves: $p_1 \neq p_2$
  - Same type: p_1 and p_2 are both double-service or both single-service
  - No missing ability $c^{p_1}_1$, $c^{p_2}_1$, $c^{p_1}_2$, and $c^{p_2}_2$ have the ability for the services assigned to them by the move
- *Effects*: patient p_1 is moved to position i^p_2 and patient p_2 is moved to position i^p_1; if $CS = \mathsf{T}$, then $c^{p_2}_1$ and $c^{p_2}_2$ are assigned to p_1, and $c^{p_1}_1$ and $c^{p_1}_2$ are assigned to p_2, for the first and second service, respectively; if $CS = \mathsf{F}$, then $c^{p_2}_2$ and $c^{p_2}_1$ are assigned to p_1 and $c^{p_1}_2$ and $c^{p_1}_1$ are assigned to p_2, for the first and second service, respectively.

Consider again the solution in Fig. 3, represented by $\Pi = [4, 3, 1, 5, 2, 6]$ and $\Theta = [(3, -), (2, -), (3, -), (1, 2), (1, 3), (1, 2)]$. The move $\mathsf{SP}\langle 5, 6, \mathsf{F}\rangle$ would lead to the state $\Pi = [4, 3, 1, \mathbf{5}, 2, \mathbf{6}]$ and $\Theta = [(3, -), (2, -), (3, -), (1, 2), (\mathbf{1}, \mathbf{2}), (\mathbf{1}, \mathbf{3})]$, where the affected values are highlighted in boldface.

4.3.3. *Neighborhood* InRouteSwap

The third neighborhood, called InRouteSwap (IRS), is also a swap move but of a different type. It swaps two patients in positions j_1 and j_2 within the route of a given caregiver c. If one or both patients are double-service ones, the route of the *side* caregiver(s) serving the patient(s) is modified accordingly. The patient is moved to the position in the route corresponding to their new global position. Differently from the SP neighborhood, side caregivers are not swapped, thus creating a different type of movement.

We call l_c the length of the route for caregiver c, that is, the number of patients served by caregiver c. We also call p_1 and p_2 the patients in positions j_1 and j_2 of the route of c, and i_1 and i_2 their global positions, that is, their positions in Π.

- *Attributes*: $\langle c, j_1, j_2\rangle$, with $c \in \mathcal{C},\ 1 \le j_1 < j_2 \le l_c$.
- *Preconditions*:
  - Minimum route length: $l_c \ge 2$
- *Effects*: Patient p_1 is moved to global position i_1 and p_2 is moved to global position i_2. The routes of c and the other caregivers of patients p_1 and p_2 are updated accordingly.

Consider once again the solution in Fig. 3, represented by $\Pi = [4, 3, 1, 5,\ 2,6]$ and $\Theta = [(3, -), (2, -), (3, -), (1, 2), (1, 3), (1, 2)]$. The move $\mathsf{IRS}\langle 3, 2, 3\rangle$ swaps the second and the third patients in the route of c_3. The new ordering obtained is $\Pi = [4, 3, \mathbf{5}, 1, 2, 6]$ (where the affected value is highlighted in boldface as before), and Θ is unchanged. Notice that p_5 is a double-service patient also served by c_1, so the route of c_1 could be affected. In this case, however, given that the new position of p_5 is remained within the positions of p_4 and p_6, the route of c_1 is unchanged.

4.3.4. *Composition of neighborhoods*

The neighborhood used in our solution method is $\mathsf{MP} \cup \mathsf{SP} \cup \mathsf{IRS}$, that is, the set union of the three basic ones defined above. In particular, a random move from the union neighborhood is obtained by first selecting one of the three basic ones and then drawing a random move within that basic neighborhood.

The first selection is a weighted randomization that makes use of two parameters, called σ_{SP} and σ_{IRS}, so moves of type SP and IRS are drawn with probability σ_{SP} and σ_{IRS}, respectively. Consequently, we draw an MP move with probability $1 - \sigma_{\mathsf{SP}} - \sigma_{\mathsf{IRS}}$. Within a single neighborhood, the specific move is selected uniformly.

It is worth mentioning that the MovePatient neighborhood is the most important one, as confirmed later by the experimental analysis. It is easy to verify that the search space is connected under this neighborhood, as it is possible to go from any solution to any other one through a chain of MovePatient moves. This is not true for the other two neighborhoods, which do not alter the route length. Instead, these can be viewed as auxiliary neighborhoods that contribute to diversifying the search.

We also notice that there is an overlap between SwapPatients and InRouteSwap, as a move that swaps two single service patients on the same route can be obtained from both of them. Given that we do not explore the neighborhood exhaustively but rather sample random moves this is not a source of inefficiency, but only a (slight) bias in the distribution.

4.4. Simulated annealing

To guide the local search, we employ SA as introduced by Kirkpatrick et al. (1983). For a detailed review of the various variants of SA, we recommend the comprehensive study conducted by Franzin and Stützle (2019).

The SA procedure starts with an initial solution constructed as outlined in Section 4.2. Subsequently, a random move within the neighborhood is selected at each iteration.

The acceptance of a move is determined based on the difference in cost, denoted as Δ, between the current solution and the new solution. The move is always accepted if Δ is negative or zero, indicating an improvement or equal value in the objective function. However, if $\Delta > 0$, the acceptance follows the Metropolis criterion, where the move is accepted with a probability of $e^{-\Delta/T}$. Here, T represents a control parameter known as the *temperature*.

The SA algorithm starts with an initial temperature denoted T_0. Subsequently, the temperature is reduced following the classical geometric cooling scheme, with $T_i = \alpha \cdot T_{i-1}$, after generating a fixed number of samples N_s. To accelerate the early stages of the search, we incorporate a *cutoff* mechanism, which causes the temperature to decrease even if the maximum number of accepted moves has been reached. Specifically, a fraction ρ of the total number of iterations N_s, where $0 \leq \rho \leq 1$, determines the reduction in temperature. The iterations saved by the cutoff mechanism are evenly distributed across all subsequent temperature levels.

To guarantee the same running time for all configurations of SA, we use the total number of iterations $\mathcal{I}$ as the stop criterion. To keep $\mathcal{I}$ fixed, we recompute N_s from $N_s = \mathcal{I} \Big/ \left(\frac{\log(T_f/T_0)}{\log \alpha} \right)$, where T_f is the final temperature.

5. Datasets and generators

In this section, we first describe the datasets already available for this problem. Subsequently, we introduce our generator and describe the new dataset obtained using our generator. Finally, we discuss the file formats for instances and solutions.

5.1. Dataset by Mankowska et al

The instances of Mankowska et al. (2014) are artificial and created by a generator, which samples random locations in a square of size 100×100 and assigns Euclidean (and thus symmetric) distances among them. The number of double-service patients is fixed to be around 30% of the total number of patients, equally distributed between those requiring simultaneous and sequential services.

Table 3
Summary of features of the different instance groups.

Group	P	S	C	Double service	Time windows	Compatibility min–max	Distance min–max
Mankowska et al. (2014)							
A	10	6	3	30%	120	0.33−0.56	39.6−50.1
B	25	6	5	32%	120	0.35−0.52	38.4−45.9
C	50	6	10	30%	120	0.34−0.49	39.5−45.0
D	75	6	15	31%	120	0.33−0.46	39.3−44.7
E	100	6	20	30%	120	0.37−0.43	39.4−44.7
F	200	6	30	30%	120	0.38−0.46	38.9−42.0
G	300	6	40	33%	120	0.37−0.42	39.2−42.1
Kummer (2021)							
I–10	10	6	3	40%	120	0.33−0.67	7.8−27.6
I–25	25	6	5	32%	120	0.20−0.60	15.2−23.2
I–50	50	6	10	32%	120	0.10−0.50	11.9−24.9
I–75	75	6	15	32%	120	0.07−0.47	10.9−24.8
I–100	100	6	20	32%	120	0.10−0.50	11.4−22.6
I–200	200	6	40	30%	120	0.20−0.48	12.7−22.4
I–300	300	6	60	30%	120	0.22−0.43	12.3−22.1
I–400	400	6	80	30%	120	0.24−0.43	11.1−22.4

The number of services is fixed globally to the value of 6, and their duration is a random number between 10 and 20, and it is identical for all patients/caregivers within the instance.

The dataset comprises seven groups, each containing ten instances with varying patient counts: 10, 25, 50, 75, 100, 200, and 300 patients, respectively. These groups are denoted by the letters from A to G, resulting in instance names such as $A_1, A_2, \ldots, G_{10}$.

The upper section of Table 3 displays the primary characteristics of this dataset, encompassing the number of patients (P), services (S), caregivers (C), the percentage of double-service patients, the duration of time windows, and the minimum and maximum average compatibility between patients and caregivers (per service). Moreover, it includes the minimum and maximum average distances in minutes between patients and the central office.

It is worth noting that instances within different groups exhibit considerable homogeneity. This uniformity stems from the fact that they were generated using the same generator configuration.

5.2. Dataset by Kummer

Kummer (2021) analyzed the instances by Mankowska et al. (2014), highlighting that they are somewhat unrealistic due to the lack of “structure” in the geographic data. For this reason, he developed a new generator that overcomes this weakness by sampling points in real cities, matching them with accurate addresses, and computing actual routes. As a consequence, distances are not Euclidean and not symmetric. However, the ratio between different patient types is fixed exactly as in the dataset by Mankowska et al. (2014), and the number of services is fixed to six as well. Nonetheless, the distribution of the abilities is more balanced to provide against the possibility that some services are assigned to a tiny number of caregivers.

To generate the geographic data of the instances, Kummer (2021) identified three features: node generation strategy (random or clustered), central office placement strategy, and cluster density. He selected 22 different combinations of values for these features and generated 100 instances for each combination. The groups' size values are the same seven of Mankowska et al. (2014), plus an additional size of 400 patients, for a total of eight groups.

As a result, he generated 17,600 (22×100×8) instances that are publicly available at https://github.com/afkummer/hhcrsp-dataset-2021 and can be used for training. The main features of this dataset are shown in the lower part of Table 3.

Inside this dataset, the author identified 160 instances (20 for each group) as the validation ones. Within each group, the set of 20 instances is composed of the 10 hardest ones and 10 random ones. The hardness of an instance is measured as the difference between the lower bound and the best-obtained value within a given time limit. The lower bounds are computed by relaxing synchronization constraints and time windows from the MIP model implemented using CPLEX.

5.3. Our generator and dataset

The generator of Kummer (2021) is surely an improvement over the original one by Mankowska et al. (2014), but it does not consider the real distribution of population in the city.

We propose to create a realistic instance generator that utilizes the actual population distribution and actual road distances. It relies on the assumption that the density of patients in a given area is directly proportional to the population density.

The generator uses the GEOSTAT cartographic database provided by the Joint Research Centre and DG Regional Policy of Eurostat. The database partitions the entire area of the European Union and some adjacent countries into square cells measuring 1 km^2. Each cell in the database contains information regarding the residing population.

Additional factors, such as age distribution and income level, which could potentially influence the demand for and the accessibility of homecare services, are not included in the GEOSTAT database; therefore, we could not use them in our generator. However, in principle, the model can be adapted to incorporate such factors if the relevant data are available.

We selected various areas, including urban, rural, and mountainous regions, with varying features concerning population, morphology, urban sprawl, and compactness. This diverse selection was made to create a wide range of instances. Within each chosen area, we sample N points, plus an additional point for the central office, specified by their latitude and longitude coordinates. The sampling process is weighted according to the population of each cell, meaning that points in cells with higher populations are more likely to be selected. Afterward, the resulting $N + 1$ points are matched to the nearest location accessible by road or street. This step is essential for calculating the road distances between pairs of points. We employ the Open Source Routing Machine (OSRM) to compute the time it takes to travel by car from one point u to another v. The distances (u, v) and (v, u) are computed independently. The result is an asymmetric complete graph where the $N + 1$ vertices represent the N patients' homes and the central office. The graph's edges are weighted with actual road travel times as computed offline by the OSRM routing engine.

We generated a dataset composed of 200 instances sampling various Italian territories of different sizes and population densities. The values of the main features, such as patients, caregivers, and services, are selected randomly for each instance (so that there are no groups).

Table 4
Features of new validation instances

Instance	*P*	*S*	*C*	Double service	Time window avg	Compatibility min–max	Distances avg	Radius (km)
0	299	8	42	6.7	124.7	0.36−0.40	42.20	26
1	165	4	24	41.8	117.7	0.42−0.58	23.61	19
2	229	8	33	47.6	124.6	0.30−0.52	17.55	11
3	44	8	8	43.2	106.4	0.25−0.50	22.58	19
4	258	4	30	17.8	117.6	0.47−0.53	33.70	26
5	212	4	28	16.0	120.0	0.39−0.61	39.42	29
6	213	8	30	28.2	120.6	0.33−0.47	43.32	30
7	297	4	45	9.8	116.4	0.49−0.51	39.53	29
8	247	8	36	32.4	127.4	0.33−0.39	33.37	32
9	55	4	7	27.3	115.9	0.43−0.57	21.55	15
10	76	4	10	30.3	120.2	0.40−0.60	20.82	15
11	181	8	27	16.0	125.1	0.22−0.52	34.48	33
12	130	4	21	24.6	123.7	0.38−0.62	38.09	37
13	131	8	18	35.1	124.7	0.22−0.44	23.95	15
14	213	4	33	42.3	125.4	0.42−0.58	35.61	29
15	73	4	11	34.2	114.7	0.36−0.64	21.50	15
16	145	4	15	13.8	116.7	0.47−0.53	21.18	11
17	101	4	13	11.9	118.1	0.38−0.62	24.96	26
18	356	4	51	42.7	126.3	0.37−0.63	24.09	17
19	203	8	28	42.9	126.1	0.29−0.43	24.94	18
20	78	4	12	46.2	127.7	0.33−0.67	20.10	15
21	323	4	49	30.3	123.7	0.35−0.65	30.86	26
22	255	4	37	25.9	114.5	0.38−0.62	36.66	29
23	75	4	9	30.7	123.8	0.44−0.56	18.34	15
24	270	4	33	11.9	123.3	0.36−0.64	30.09	25
25	45	8	8	28.9	113.7	0.13−0.63	22.82	18
26	191	4	26	18.9	124.5	0.38−0.62	25.83	17
27	157	8	29	47.8	125.7	0.17−0.55	16.74	10
28	378	8	51	19.0	123.0	0.33−0.41	37.70	32
29	100	4	11	3.0	115.5	0.45−0.55	32.87	21

We randomly selected 30 instances for validation from this dataset while retaining the remaining 170 instances for training. The features of the validation instances are presented in Table 4. The time window for each patient spans between 60 and 180 minutes. Furthermore, in addition to the features outlined in Table 3, we also provide the distance radius from the central office, which was used during the generation of the instances.

5.4. *File formats*

Unfortunately, there is no single well-established file format for this problem. Indeed, the dataset of Mankowska et al. (2014) uses two different formats, one for the instances of up to 75 patients and one for the larger ones (from 100 patients upward). Kummer (2021)'s dataset uses a third one, which is similar, but not identical to the second one by Mankowska et al. (2014)

All three formats are text only and have a fixed structure. They are easy to parse but quite fragile and not very human-readable. In addition, they contain some redundant data. For these reasons, we moved to a more robust and human-readable format based on JSON. Our JSON format is extensible to different, more complex versions of the problem, which is not the case for the previous ones. The solutions are also written in JSON, with their specific syntax.

In our repository located at https://github.com/iolab-uniud/hhcrsp , we provide a brief guide on the input and output JSON file syntax. This repository contains all the validation instances and our best solutions. Additionally, we offer a Python-based validator to ensure the correctness of instances and solutions.

6. Experimental analysis

Our solution method is implemented in C++ using the framework for local search *EasyLocal++* (Ceschia et al., 2024).

The experiments have been run on an Ubuntu Linux 22.4 machine with 4 cores Intel® i7-7700 (3.60 GHz), with a single core dedicated to each experiment.

6.1. Parameter tuning

The parameter tuning was performed using the tool JSON2RUN (Urli, 2013), which samples the configurations using the *Hammersley point set* (Hammersley and Handscomb, 1964), and implements the F-Race procedure (Birattari et al., 2010) to compare them. The F-Race procedure uses the Friedman and Wilcoxon rank-sum tests to sort out the configurations that are statistically inferior to the others.

The tuning procedure was performed in two stages. In Stage I, we tuned the parameters of SA, namely T_0, T_f, α, and ρ; in Stage II, we tuned the rates of the neighborhoods σ_*. In Stage I, we used rates obtained from preliminary experiments, whereas in Stage II we used the SA parameters obtained in Stage I.

Tuning has been performed using only the training dataset proposed by Kummer (2021).

We conducted experiments with 30 Hammersley points and identified the best result, presented in Table 5. The initial parameter ranges were determined through preliminary experiments and provided in the table. During the tuning phase, the maximum number of iterations $\mathcal{I}$ for the SA algorithm was set to 5×10^7, leading to an average running time of approximately 88 seconds on our machine.

We notice that σ_{SP} and σ_{IRS} have a rather low value in the winning configuration, both equal to 0.04. Indeed, experiments using only neighborhood MovePatient obtain results that are not much inferior to the multineighborhood setting (see Section 6.5). However, especially for the largest and more challenging instances, the contribution of the two auxiliary neighborhoods is statistically non-negligible. In fact, the configuration with $\sigma_{\mathsf{SP}} = 0$ and $\sigma_{\mathsf{IRS}} = 0$ was eliminated by the RACE procedure.

Table 5
Parameter settings

Name	Description	Value	Range
T_0	Initial temperature	6.11	3−10
T_f	Final temperature	0.13	0.05−0.15
α	Cooling rate	0.9863	0.985−0.995
ρ	Accepted moves ratio	0.063	0.05−0.15
σ_{SP}	Probability of SP	0.04	0.0 − 0.3
σ_{IRS}	Probability of IRS	0.04	0.0 − 0.3

6.2. Comparative results on the dataset of Mankowska et al

We excluded from our analysis group A as all 10 instances are always consistently solved to optimality, so they are not sufficiently challenging.

Regarding the parameter setting, due to the Metropolis acceptance criterion of SA, temperatures must be related to costs. These costs are, in turn, influenced by distances and tardiness. Given that in this dataset distances represent the biggest cost, and they are consistently larger than the distances in the training instances by Kummer (2021), we rescaled the parameters T_0 and T_f. To this aim, we observed that the average distance in this dataset is 2.3 times larger than in the training one; therefore, we multiplied T_0 and T_f by 2.3 compared to the values in Table 5, keeping all the others fixed as in the table.

The comparison of our results with the best-known results in the literature is presented in Tables 6 and 7 for different instance groups, namely B-D and E-G, respectively. Specifically, we compare our findings (3SA) with those obtained by Mankowska et al. (2014) and Lasfargeas et al. (2019) using variable neighborhood search, as well as by Kummer et al. (2020, 2024) using biased random-key genetic algorithms. Following usual conventions, the best (average) results are indicated in bold while the optimal values (only found in group B) are underlined.

We ran 10 replicates of our solving procedure for each instance, with the number of iterations set at $\mathcal{I} = 10^8$. This corresponds to approximately 6 minutes of running time for the largest instances in group G. Although this allocated time is less than what others granted for the same instances, it led to significantly longer running times (up to 2 orders of magnitude) for us in the case of smaller instances. Nevertheless, our solver exhibits more linear time scalability than our competitors.

While an entirely fair comparison is not feasible due to different running times and processors, it is evident that our best and average results outperform the others in nearly all cases. The results reinforce this observation averaged on each group, as indicated in the *Avg* rows in the tables. Furthermore, we consistently achieved optimal solutions for seven out of 10 instances in group B. We always find the same value for the remaining three, but whether it is optimal is unknown. All our results have been validated using the MIP model by Mankowska et al. (2014), implemented in CPLEX, and kindly supplied to us by Alberto Kummer.

6.3. Comparative results on the dataset of Kummer

Tables 8–10 present the comparative results for the second dataset, with group I-10 removed. As explained above, the validation dataset comprises the 10 most challenging instances and 10 random

Table 6
Comparative results on instances of groups $B - D$

	Kumm-22	Mank-14		Lasf-19			Kumm-20			Kumm-22			3SA		
Inst.	LB	cost	t (s)	best	avg	t (s)	best	avg	t (s)	best	avg	t (s)	best	avg	t (s)
B_1	428.10	458.9	<1	434.1	552.8	53.1	428.10	428.26	8.6	428.10	428.53	0.8	428.10	**428.10**	73.9
B_2	476.05	476.2	<1	476.0	561.3	27.7	483.63	485.66	8.4	476.05	476.92	0.9	476.05	**476.05**	74.2
B_3	399.09	399.2	<1	399.1	527.6	63.5	402.80	402.80	9.0	402.80	409.29	1.0	399.09	**399.09**	75.6
B_4	411.30	576.0	<1	414.0	509.7	66.8	420.29	431.87	8.2	422.06	430.46	1.1	411.30	**411.30**	74.7
B_5	366.34	391.1	<1	385.6	496.9	13.7	372.16	374.24	8.2	369.44	375.15	1.0	366.34	**366.34**	74.2
B_6	405.58	534.7	<1	447.8	611.8	443.7	471.00	471.87	8.9	470.59	470.70	1.2	464.62	**464.62**	73.9
B_7	328.67	355.5	<1	328.7	398.8	61.5	328.67	**328.67**	9.5	328.67	**328.67**	0.9	328.67	**328.67**	72.7
B_8	357.68	357.8	<1	359.7	488.7	79.3	359.70	359.70	9.2	357.68	359.40	0.7	357.68	**357.68**	73.9
B_9	330.30	403.8	<1	404.1	483.4	62.1	402.67	404.27	10.0	404.11	404.29	0.9	402.67	**402.67**	75.6
B_{10}	420.99	500.4	<1	462.7	616.8	8.7	469.58	469.58	9.2	469.58	469.58	0.9	462.75	**462.75**	73.6
Avg		445.4		411.2	524.8	88.0	413.86	415.69	8.9	412.91	415.30	0.9	409.73	**409.73**	74.24
C_1	459.25	1123.6	<1	974.2	1350.4	96.2	965.15	975.59	36.9	969.11	**973.87**	3.1	943.73	1010.79	105.7
C_2	373.94	673.8	<1	605.1	685.5	106.4	583.39	590.48	39.0	584.18	587.00	2.9	569.12	**575.51**	103.6
C_3	390.48	642.4	<1	562.9	698.2	109.8	548.79	559.05	37.4	549.63	**552.52**	2.9	537.79	563.10	105.9
C_4	371.99	580.4	<1	521.9	630.4	112.4	519.91	530.59	36.2	520.13	524.15	3.0	495.17	**499.23**	104.1
C_5	464.97	754.6	<1	683.1	822.6	114.9	678.61	702.92	31.4	668.65	685.92	3.4	655.72	**666.50**	102.8
C_6	360.73	951.6	<1	854.6	1010.6	115.9	840.69	845.49	37.2	841.48	846.83	2.9	813.25	**831.62**	106.9
C_7	354.15	577.4	<1	529.2	572.5	109.4	534.85	540.39	42.2	533.92	541.88	3.4	511.89	**515.29**	103.0
C_8	375.52	540.6	<1	471.0	522.8	110.8	474.55	480.06	36.5	475.96	478.39	3.5	468.88	**470.26**	103.1
C_9	355.29	608.7	<1	551.1	642.7	115.4	534.30	551.14	42.8	545.18	558.54	3.4	527.69	**535.63**	105.1
C_{10}	431.18	679.3	<1	608.9	653.0	99.0	611.25	618.27	35.3	611.03	614.59	2.7	590.26	**590.27**	103.4
Avg		713.2		636.2	758.9	109.0	629.15	639.40	37.5	629.93	636.37	3.1	611.35	**625.82**	104.36
D_1	492.09	1321.8	5	1278.2	1498.8	143.0	1186.20	1209.62	93.7	1193.21	1215.79	8.0	1110.81	**1182.21**	136.4
D_2	384.68	892.7	4	746.9	914.3	168.7	693.28	718.03	82.9	679.58	695.99	7.2	653.73	**673.26**	134.7
D_3	380.05	819.4	4	678.6	817.8	155.4	635.67	651.35	102.2	644.16	650.22	8.5	613.12	**634.94**	135.4
D_4	418.94	877.4	4	809.7	1073.1	148.5	814.35	841.64	82.3	795.15	827.28	7.2	770.60	**784.07**	134.9
D_5	415.81	872.1	5	777.0	924.9	150.3	691.50	703.12	92.4	693.83	702.68	7.7	651.65	**661.18**	135.4
D_6	392.08	835.2	5	768.6	886.6	154.6	733.67	744.70	105.8	731.71	743.64	7.9	688.15	**692.65**	134.9
D_7	372.49	706.3	6	600.1	680.4	168.1	590.64	604.75	112.8	586.10	597.25	7.8	565.78	**580.54**	133.0
D_8	409.35	811.4	4	715.5	775.8	149.8	661.78	680.31	102.7	658.49	669.83	8.1	647.95	**662.06**	133.3
D_9	385.89	842.7	6	741.0	818.2	156.0	706.08	723.45	92.6	689.83	710.32	9.2	651.10	**660.49**	134.8
D_{10}	485.63	1306.6	3	1424.6	1867.7	173.1	1208.71	1290.56	77.7	1189.32	1280.92	6.9	1155.89	**1165.73**	136.3
Avg		928.6	4.6	854.0	1025.8	156.8	792.19	816.75	94.5	786.14	809.39	7.9	750.88	**769.71**	134.91

ones out of 2200 generated for each group. The most challenging instances in our tables are on the left, and the random ones are on the right. The comparison here is only against the results of Kummer (2021) as these instances have been proposed recently and are not considered by other studies.

We can see that our results outperform the previous ones in most instances, particularly the largest ones. The only group in which the previous results are cumulatively superior is the smallest one (25 patients) of hard instances.

As for the previous dataset, our running times, corresponding to 10^8 iterations, are longer than Kummer (2021) for small instances and shorter for large ones.

Table 7
Comparative results on instances of groups $E - G$

	Kumm-22	Mank-14		Kumm-20			Kumm-22			3SA		
Inst.	LB	cost	t (s)	best	avg	t (s)	best	avg	t (s)	best	avg	t (s)
E_1	430.36	1604.9	17	1331.49	1352.32	193.6	1327.72	**1340.36**	17.2	1257.71	1357.71	166.1
E_2	444.88	1101.9	10	848.08	871.25	192.0	829.79	865.05	17.1	780.57	**791.59**	163.4
E_3	454.27	986.4	14	788.03	814.23	182.9	789.56	806.53	16.7	758.24	**775.46**	165.9
E_4	412.08	871.0	19	711.19	729.93	196.5	723.87	728.96	14.8	679.57	**696.71**	166.8
E_5	416.62	1018.0	19	781.50	803.86	182.3	780.04	817.13	17.0	714.00	**739.40**	164.9
E_6	416.60	1003.0	19	790.47	804.16	177.6	779.82	793.68	18.3	751.26	**762.77**	167.0
E_7	389.57	921.1	20	711.11	733.91	191.8	705.79	715.46	18.0	680.55	**692.67**	165.4
E_8	433.89	884.6	19	752.35	761.97	168.8	733.90	750.59	17.2	708.08	**717.51**	167.3
E_9	446.49	1131.7	18	921.78	951.91	163.1	893.35	916.56	16.4	840.14	**869.21**	166.2
E_{10}	455.07	1053.6	11	825.24	845.10	174.5	822.85	841.57	16.0	782.76	**802.24**	163.9
Avg		1057.6	16.6	846.12	866.86	182.3	838.67	857.59	16.9	795.29	**820.53**	165.69
F_1	548.88	1721.4	889	1401.96	1425.97	745.5	1311.10	1351.20	124.4	1229.68	**1274.23**	298.8
F_2	543.32	1763.8	909	1336.33	1383.58	812.1	1298.31	1337.41	121.7	1214.39	**1248.10**	296.8
F_3	547.64	1549.6	868	1263.39	1285.81	780.3	1215.96	1272.23	116.5	1136.95	**1170.20**	298.0
F_4	531.84	1420.4	1321	1124.24	1146.22	901.9	1100.66	1134.66	136.0	1027.68	**1070.69**	297.4
F_5	538.14	1701.9	1145	1329.29	1365.17	826.1	1298.55	1331.09	119.8	1214.61	**1240.69**	298.7
F_6	518.47	1639.7	836	1332.14	1373.32	649.8	1292.52	1368.41	109.6	1231.47	**1264.24**	300.6
F_7	512.98	1384.3	1294	1131.27	1157.35	817.0	1084.57	1125.37	120.5	1063.33	**1093.83**	296.4
F_8	536.15	1544.6	924	1132.77	1165.15	716.4	1123.22	1140.42	107.7	1077.12	**1109.37**	298.5
F_9	543.16	1572.9	1642	1311.43	1345.01	770.4	1263.19	1344.62	125.4	1183.69	**1215.89**	299.3
F_{10}	546.84	1581.0	1326	1418.53	1446.35	740.3	1383.08	1419.76	119.6	1274.71	**1306.27**	301.0
Avg		1588.0	1115.4	1278.14	1309.39	776.0	1237.12	1282.52	120.1	1165.36	**1199.35**	298.55
G_1	612.37	2248.0	7200	1778.54	1855.17	1949.8	1744.14	1824.34	439.4	1668.14	**1707.70**	456.7
G_2	605.84	2316.1	7200	1824.74	1897.99	2115.1	1709.70	1799.78	519.5	1633.70	**1668.64**	454.0
G_3	614.20	1885.3	7147	1514.23	1546.53	1935.1	1464.69	1511.86	461.6	1387.44	**1438.69**	453.8
G_4	604.30	2023.2	7200	1564.42	1599.39	2137.6	1508.94	1569.01	529.0	1448.65	**1491.84**	455.3
G_5	633.66	2247.6	7200	1698.28	1749.10	1840.9	1652.88	1681.01	466.6	1547.88	**1585.38**	458.3
G_6	621.46	2144.4	7200	1714.38	1777.39	2014.4	1681.64	1719.18	570.5	1630.98	**1680.26**	457.6
G_7	602.42	1971.5	6934	1640.07	1677.92	1844.3	1536.00	1604.96	522.4	1494.00	**1534.86**	453.2
G_8	618.74	1987.4	7200	1547.63	1583.86	1799.1	1498.38	1535.90	531.7	1444.88	**1478.84**	453.3
G_9	662.70	2415.5	7023	1942.21	1972.48	1810.7	1850.07	1976.27	446.6	1762.51	**1799.20**	457.9
G_{10}	633.76	2373.4	7003	1872.08	1932.27	1649.6	1785.37	1868.56	482.8	1664.70	**1714.07**	456.9
Avg		2161.2	7130.7	1709.66	1759.21	1909.7	1643.18	1709.09	497.0	1568.29	**1609.95**	455.7

6.4. Results on our new dataset

We now show the results obtained for the validation instances of our new dataset. We have no comparative results available in this case, so we only show our values for future comparisons. We report in Table 11 the average for 10 runs of the cost (obtained using the three weights equal to 1/3) and the individual cost components (distance, total tardiness, highest tardiness). To give a more qualitative view of the costs, we also show the average distance per caregiver and tardiness per provided service. The last row shows the averages for all instances of these measures.

The average distance (in minutes) for caregivers is 122.8, meaning that an operator spends about 2 hours of their working day traveling between patients and the central office. Average tardiness is

Table 8
Comparative results on instances of groups 25 – 50

	Kumm-21			3SA				Kumm-21			3SA		
Inst.	best	avg	t (s)	best	avg	t (s)	Inst.	best	avg	t (s)	best	avg	t (s)
25_5_22_1.0_C_C	820.33	**831.47**	49.07	820.33	862.68	69.33	25_5_80_0.8_C_RC	436.67	**436.67**	49.07	436.33	443.13	71.13
25_5_69_1.6_R_RC	1149.00	**1149.00**	49.01	1146.00	1162.92	68.05	25_5_48_1.0_C_C	275.00	277.18	49.53	272.0	**272.19**	71.66
25_5_47_0.8_R_C	848.00	**848.00**	49.15	842.33	850.07	71.61	25_5_42_0.8_C_RC	287.67	**287.67**	49.41	287.33	288.70	72.88
25_5_37_0.8_R_RC	1434.67	**1434.67**	49.23	1434.67	1442.24	71.04	25_5_56_0.8_R_RC	178.33	**178.80**	50.09	177.33	179.29	72.33
25_5_37_1.0_R_RC	1434.67	**1434.67**	49.15	1434.67	1442.47	70.98	25_5_76_1.2_C_C	183.67	**183.67**	49.65	183.67	185.27	71.44
25_5_26_0.8_C_C	956.33	**956.33**	49.20	952.33	996.70	71.72	25_5_70_1.0_R_C	146.00	146.00	50.02	141.67	**142.23**	70.67
25_5_6_0.8_R_C	681.00	687.60	49.55	678.67	**682.94**	72.03	25_5_21_1.6_R_RC	204.67	204.67	50.04	197.67	**200.20**	68.81
25_5_69_1.6_R_C	695.00	**696.06**	49.06	695.00	724.22	69.57	25_5_73_0.4_R_RC	151.00	152.85	49.59	150.00	**150.00**	70.92
25_5_42_0.8_R_RC	1161.67	1168.45	48.98	1158.33	**1165.91**	69.98	25_5_97_1.0_R_R	258.67	**258.67**	49.41	258.67	**258.67**	71.36
25_5_96_0.8_R_C	707.67	**707.67**	49.49	697.67	718.98	72.68	25_5_49_0.4_C_C	138.67	138.69	49.86	137.67	**137.67**	71.79
Avg.	988.83	**991.39**	49.19	986.00	1004.91	70.70	Avg.	226.04	226.49	49.66	224.23	**225.74**	71.30
50_10_80_1.0_C_C	826.67	829.05	58.27	815.67	**818.54**	102.56	50_10_3_1.6_C_C	262.00	266.25	61.58	251.0	**252.83**	99.30
50_10_81_1.0_R_RC	999.67	1010.04	58.12	986.33	**994.50**	100.58	50_10_53_1.2_R_C	250.67	254.92	60.59	247.0	**251.96**	101.15
50_10_80_1.0_C_RC	1057.33	1063.25	58.28	1045.0	**1048.74**	103.16	50_10_62_1.2_R_C	211.33	218.25	62.81	204.0	**205.80**	100.40
50_10_97_0.8_C_RC	759.67	**770.72**	59.21	749.67	779.74	102.47	50_10_74_1.0_R_RC	290.33	**291.93**	60.31	289.67	297.64	101.22
50_10_32_1.2_R_C	442.33	463.30	60.76	431.33	**454.56**	101.79	50_10_61_1.0_R_RC	276.33	279.62	62.71	270.0	**271.57**	100.68
50_10_3_0.8_R_C	589.00	596.18	60.90	585.67	**589.31**	101.19	50_10_88_1.6_C_RC	317.67	320.93	61.92	308.0	**312.92**	100.25
50_10_80_0.4_C_C	762.67	762.84	58.77	752.33	**757.70**	101.40	50_10_60_0.4_C_RC	266.33	267.38	62.30	261.0	**265.28**	99.99
50_10_26_1.0_C_RC	610.33	**611.10**	59.83	607.0	612.78	102.94	50_10_2_1.0_C_R	345.00	348.70	60.77	342.33	**346.11**	100.90
50_10_26_1.2_R_C	312.00	317.02	60.91	303.0	**310.86**	100.46	50_10_100_1.6_C_C	159.00	159.00	61.92	154.67	**156.13**	99.73
50_10_44_1.6_R_C	279.33	284.55	62.56	270.0	**279.44**	99.97	50_10_33_0.4_R_RC	279.67	280.64	62.12	274.33	**280.54**	100.56
Avg.	663.90	670.80	59.76	654.60	**664.62**	101.65	Avg.	265.83	268.76	61.70	260.20	**264.08**	100.42

Table 9
Comparative results on instances of groups 75–100.

	Kumm-21			3SA				Kumm-21			3SA		
Inst.	best	avg	*t* (s)	best	avg	*t* (s)	Inst.	best	avg	*t* (s)	best	avg	*t* (s)
75_15_97_1.0_R_RC	798.00	814.52	72.50	779.67	**806.37**	131.48	75_15_97_0.4_C_C	480.00	483.37	74.83	462.0	**470.77**	131.27
75_15_35_0.8_R_C	719.00	730.60	72.90	704.0	**717.74**	130.02	75_15_63_1.0_R_RC	408.00	413.10	78.22	395.0	**400.63**	128.86
75_15_35_1.0_C_RC	1051.33	1058.83	73.27	1036.0	**1046.3**	131.12	75_15_5_0.8_C_RC	439.67	446.17	78.09	426.67	**439.03**	129.83
75_15_97_1.0_C_C	612.33	635.02	74.86	593.0	**604.67**	133.54	75_15_19_0.8_R_C	368.67	377.88	77.86	361.67	**373.88**	129.25
75_15_79_1.2_R_C	404.33	421.37	79.09	394.33	**400.69**	126.11	75_15_84_0.8_R_RC	392.33	401.47	76.96	381.67	**389.83**	128.59
75_15_10_1.6_R_C	376.00	390.67	74.60	343.0	**357.67**	128.95	75_15_30_1.0_R_C	303.67	309.13	80.97	292.0	**297.77**	128.82
75_15_88_1.6_R_C	461.33	480.05	79.34	449.67	**456.69**	129.86	75_15_32_1.2_R_RC	353.67	357.68	76.31	338.67	**344.21**	130.36
75_15_98_1.6_R_C	375.67	393.70	76.71	359.33	**373.92**	129.69	75_15_40_0.8_R_RC	423.67	429.10	82.88	409.33	**418.30**	129.89
75_15_21_1.6_R_C	384.67	389.02	81.12	363.67	**379.66**	126.79	75_15_63_0.4_R_C	317.33	322.50	78.69	303.67	**311.06**	130.30
75_15_87_1.6_C_C	386.33	388.57	76.80	358.67	**372.51**	129.78	75_15_65_1.0_C_C	257.00	259.53	78.81	252.0	**256.60**	128.89
Avg.	556.90	570.23	76.12	538.13	**551.62**	129.73	Avg.	374.40	379.99	78.36	362.27	**370.21**	129.61
100_20_8_1.2_R_C	607.33	**635.73**	105.73	606.67	636.84	155.84	100_20_68_1.6_R_C	304.67	313.48	115.38	306.00	**310.07**	155.21
100_20_63_1.2_R_C	548.33	**570.03**	107.07	552.00	577.51	152.74	100_20_49_0.4_C_C	525.67	536.42	103.04	512.33	**526.34**	157.46
100_20_39_1.6_R_C	554.33	567.50	99.28	540.67	**554.50**	154.97	100_20_61_0.4_R_RC	471.33	477.83	110.32	450.00	**460.97**	157.76
100_20_76_1.6_R_C	500.00	510.58	98.36	476.00	**490.37**	153.07	100_20_88_1.2_C_C	269.67	273.62	99.41	263.33	**268.83**	158.00
100_20_57_1.6_R_C	505.67	523.40	105.03	488.00	**500.22**	156.40	100_20_9_1.0_R_R	554.67	564.48	110.90	545.00	**557.51**	158.10
100_20_25_0.8_R_C	654.67	670.80	97.75	621.33	**641.31**	155.36	100_20_35_1.6_C_RC	442.67	453.77	103.82	422.00	**434.11**	157.97
100_20_77_1.0_R_C	613.67	640.15	105.58	606.67	**623.92**	152.95	100_20_10_1.2_R_C	300.00	308.20	103.62	297.67	**302.99**	158.87
100_20_11_1.2_R_C	535.33	551.07	95.21	504.67	**522.10**	156.48	100_20_65_0.4_R_RC	495.00	503.87	104.77	481.33	**491.31**	159.08
100_20_30_1.0_R_C	621.67	635.85	101.69	604.33	**629.99**	153.57	100_20_99_0.4_C_RC	487.00	492.63	107.93	469.33	**476.18**	155.73
100_20_23_1.2_R_C	454.00	467.57	107.64	438.33	**446.83**	153.67	100_20_54_1.2_R_RC	398.67	410.53	101.42	399.67	**408.83**	157.94
Avg.	559.50	577.27	102.33	543.87	**562.36**	154.51	Avg.	424.94	433.48	106.06	414.67	**423.71**	157.61

Table 10
Comparative results on instances of groups 200–400

	Kumm-21			3SA				Kumm-21			3SA		
Inst.	best	avg	*t* (s)	best	avg	*t* (s)	Inst.	best	avg	*t* (s)	best	avg	*t* (s)
200_40_52_1.6_R_C	932.00	972.72	276.99	879.67	**908.91**	289.15	200_40_17_1.6_C_C	569.67	577.58	343.10	545.0	**559.03**	289.65
200_40_7_1.2_R_C	1081.33	1113.22	263.19	957.0	**1005.39**	289.61	200_40_75_1.6_R_C	679.00	693.62	296.38	648.33	**662.33**	293.56
200_40_30_1.6_R_C	923.67	953.12	280.84	878.67	**917.73**	287.55	200_40_80_1.0_C_C	635.33	646.52	372.74	623.67	**639.77**	293.87
200_40_76_1.2_R_C	1133.33	1194.73	301.42	1081.0	**1111.1**	288.66	200_40_82_1.0_C_R	944.00	977.73	296.24	901.33	**925.96**	292.77
200_40_7_1.6_R_C	847.67	876.15	263.28	796.0	**814.70**	293.98	200_40_69_1.0_C_R	991.67	1015.40	341.85	949.0	**961.77**	295.80
200_40_51_1.0_R_C	1136.67	1206.20	292.73	1072.67	**1097.72**	287.91	200_40_60_0.4_C_RC	870.67	900.08	268.88	835.0	**853.31**	292.97
200_40_17_1.6_R_C	835.00	858.85	340.20	791.67	**810.12**	286.58	200_40_10_1.6_C_RC	843.33	859.15	323.51	809.0	**827.16**	291.85
200_40_40_1.0_R_C	913.00	942.42	297.18	863.0	**886.16**	288.03	200_40_98_1.6_C_RC	830.33	844.53	311.00	797.33	**812.43**	294.15
200_40_71_1.6_R_C	857.67	883.70	274.77	811.33	**828.67**	290.08	200_40_100_0.4_C_C	711.67	726.58	342.13	688.67	**700.76**	293.94
200_40_98_1.2_R_C	936.00	973.03	314.45	926.0	**944.30**	287.59	200_40_71_1.6_C_C	458.67	468.78	298.01	437.0	**450.84**	297.45
Avg.	959.63	997.41	290.50	905.70	**932.48**	288.91	Avg.	753.43	771.00	319.38	723.43	**739.34**	293.60
300_60_4_1.6_R_C	1568.67	1629.28	881.04	1479.33	**1515.52**	440.05	300_60_7_1.2_R_RC	1434.33	1486.10	627.71	1353.67	**1386.63**	451.88
300_60_89_1.6_R_C	1431.33	1507.85	672.63	1382.67	**1414.86**	442.91	300_60_75_1.2_C_RC	1088.33	1134.57	762.75	1057.33	**1080.89**	459.97
300_60_82_1.2_R_C	1499.67	1586.38	729.01	1428.67	**1457.96**	443.74	300_60_26_0.4_C_RC	1284.67	1343.53	689.58	1248.67	**1268.22**	455.09
300_60_95_1.2_R_C	1596.67	1696.10	845.22	1506.33	**1555.93**	443.89	300_60_68_0.8_C_RC	1182.67	1230.22	847.66	1174.33	**1193.30**	457.47
300_60_22_1.2_R_C	1458.67	1522.58	823.69	1430.0	**1461.66**	443.82	300_60_4_1.0_R_C	1016.00	1078.32	890.83	966.67	**990.40**	454.24
300_60_85_1.2_R_C	1412.33	1479.60	715.23	1307.67	**1345.38**	447.29	300_60_73_0.8_R_RC	1175.67	1234.60	740.73	1082.67	**1117.81**	456.54
300_60_81_1.6_R_C	1230.67	1313.85	785.84	1185.0	**1204.56**	447.06	300_60_45_1.6_C_C	726.00	747.50	834.72	688.67	**698.48**	457.39
300_60_23_1.2_R_C	1271.33	1322.22	823.18	1218.0	**1251.61**	451.73	300_60_46_0.4_R_C	1068.00	1113.27	801.14	1041.67	**1060.11**	460.30
300_60_34_1.2_R_C	1635.00	1709.00	801.04	1451.0	**1481.31**	445.53	300_60_81_0.8_C_C	814.00	835.68	806.91	778.33	**796.78**	461.93
300_60_69_1.0_R_C	1530.00	1579.32	789.10	1410.0	**1439.10**	444.43	300_60_48_0.8_C_C	826.00	850.58	811.38	812.33	**825.24**	464.62
Avg.	1463.43	1534.62	786.60	1379.87	**1412.79**	445.05	Avg.	1061.57	1105.44	781.34	1020.43	**1041.79**	457.94
400_80_59_1.0_R_C	2400.67	2464.35	1602.10	2135.67	**2174.41**	623.95	400_80_54_1.2_C_RC	1562.33	1675.90	1757.88	1522.67	**1542.97**	629.73
400_80_16_1.2_R_C	2201.33	2311.37	1631.95	1962.33	**1995.44**	614.56	400_80_20_0.4_C_RC	1636.33	1731.70	1430.20	1581.67	**1612.38**	641.43
400_80_51_1.0_R_C	2237.33	2425.50	1522.17	2074.67	**2126.64**	612.96	400_80_61_0.4_C_RC	1625.00	1666.07	1627.19	1556.33	**1584.70**	633.24
400_80_97_1.6_R_C	1759.67	1820.27	1733.84	1561.0	**1607.43**	618.26	400_80_33_1.0_C_RC	1527.33	1620.13	1718.09	1478.33	**1501.32**	641.89
400_80_30_1.6_R_C	1834.00	1908.13	1474.27	1678.0	**1700.80**	612.68	400_80_95_1.0_C_R	1547.33	1600.53	1660.34	1470.0	**1504.36**	638.15
400_80_34_1.0_R_C	2064.67	2145.38	1605.12	1835.33	**1880.71**	612.48	400_80_65_1.0_C_R	1618.00	1689.10	1650.58	1504.0	**1531.89**	637.37
400_80_92_1.6_R_C	1791.33	1851.02	1495.99	1599.33	**1628.06**	625.25	400_80_45_0.4_C_C	1265.67	1324.67	1526.54	1186.0	**1216.59**	638.65
400_80_91_1.2_R_C	1909.33	2025.52	1608.22	1826.33	**1864.77**	616.77	400_80_22_1.6_C_C	948.00	1001.80	1725.22	869.0	**887.37**	643.22
400_80_28_1.2_R_C	2079.00	2172.27	1553.66	1913.67	**1943.40**	621.14	400_80_77_1.2_R_RC	1447.33	1504.32	1904.06	1298.33	**1330.06**	638.64
400_80_81_1.6_R_C	1743.00	1815.32	1710.40	1600.0	**1624.63**	612.49	400_80_60_0.8_C_C	1065.67	1128.53	1413.34	1030.33	**1051.96**	645.96
Avg.	2002.03	2093.91	1593.77	1818.63	**1854.63**	617.05	Avg.	1424.30	1494.28	1641.34	1349.67	**1376.36**	638.83

Table 11
Results on the new instances

Instance	Cost	Distance	Distance per caregiver	Total tardiness	Tardiness per service	Highest tardiness
0	2058.0	6139.7	146.2	24.2	0.1	10.0
1	864.9	2583.5	107.6	7.6	0.0	3.6
2	915.6	2741.5	83.1	3.8	0.0	1.6
3	367.9	1101.8	137.7	1.1	0.0	0.8
4	1317.1	3934.5	131.2	13.4	0.0	3.3
5	1251.4	3746.1	133.8	5.6	0.0	2.4
6	1987.1	5892.9	196.4	55.4	0.2	13.0
7	1598.8	4695.8	104.4	52.5	0.2	48.0
8	1654.5	4946.2	137.4	12.4	0.0	4.8
9	301.7	902.3	128.9	1.8	0.0	1.1
10	428.4	1253.8	125.4	22.2	0.2	9.2
11	1219.7	3626.0	134.3	24.5	0.1	8.5
12	925.6	2765.3	131.7	7.6	0.0	4.0
13	693.5	2065.1	114.7	11.1	0.1	4.4
14	1327.6	3965.5	120.2	11.8	0.0	5.5
15	471.4	1274.8	115.9	106.7	1.1	32.7
16	514.4	1540.4	102.7	1.9	0.0	1.0
17	532.9	1589.3	122.3	5.6	0.0	3.7
18	1511.6	4525.5	88.7	6.4	0.0	2.8
19	1275.0	3815.9	136.3	6.6	0.0	2.5
20	447.6	1330.9	110.9	7.7	0.1	4.2
21	1897.4	5670.4	115.7	17.4	0.0	4.3
22	1644.4	4886.6	132.1	35.7	0.1	10.9
23	338.1	1009.6	112.2	3.2	0.0	1.4
24	1309.6	3910.4	118.5	13.2	0.0	5.3
25	489.5	917.9	114.7	403.7	7.0	147.0
26	1024.5	3059.0	117.7	10.7	0.0	3.8
27	620.0	1843.7	63.6	11.6	0.0	4.8
28	2249.5	6722.0	131.8	21.6	0.0	5.0
29	624.1	1847.9	168.0	18.7	0.2	5.7
avg	1062.1	3143.5	122.8	30.9	0.3	11.8

0.3 minutes, which is perfectly acceptable, although this value also considers all the services with no tardiness. The average highest tardiness is 11.8 minutes, which is relatively low and generally tolerable. We also see that there are no instances with zero tardiness, showing that this objective is indeed binding and significant.

As an example, a solution for instance 3 is shown in Fig. 4. We see only two late patients (p_1 and p_{36}), highlighted in white diagonal stripes. On the contrary, many early arrivals (e.g., p_8 and p_{14}) cause the caregivers' idleness, which is not penalized in the objective function.

6.5. *Discussion and further analyses*

In order to investigate the relative importance of the three neighborhoods, we performed an *ablation analysis* in which the two "auxiliary" neighborhoods, namely SP and IRS, are removed from the

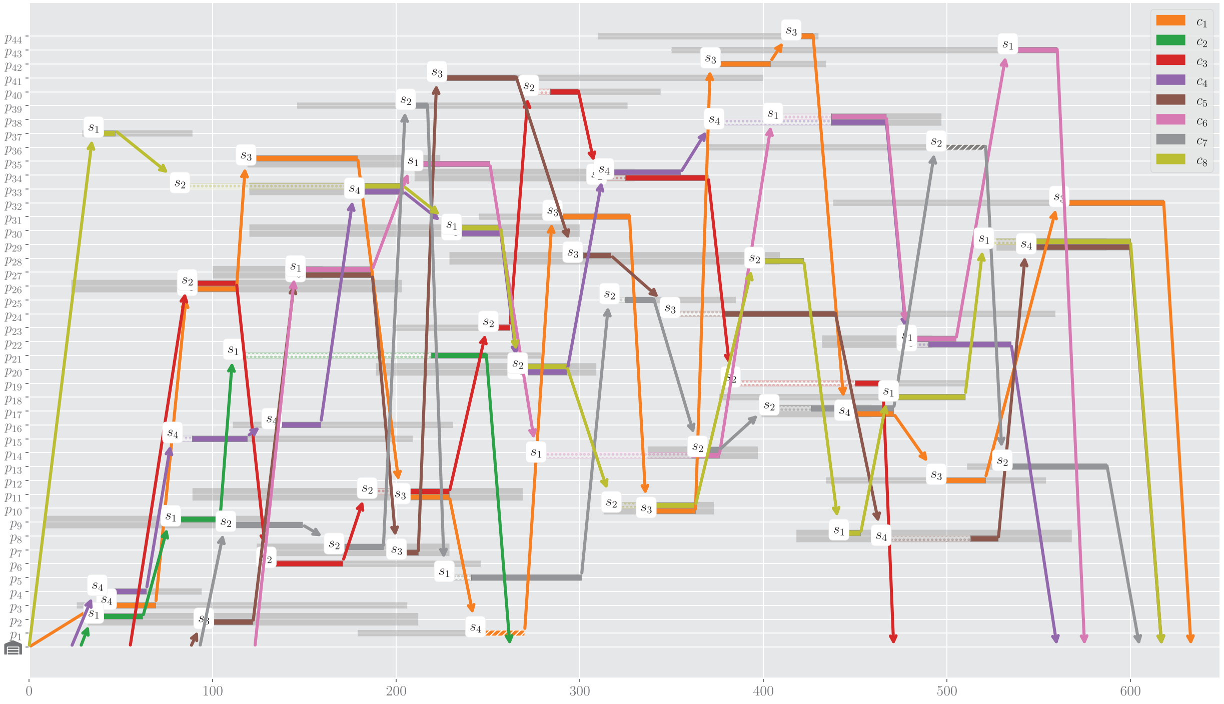

Fig. 4. Solution of instance 3.

search method one at the time. Table 12 and Fig. 5 show the percentage gap between the original method, that uses the neighborhood MP∪SP∪IRS, and the ones that use MP∪SP, MP∪IRS, and MP, respectively.

We observe that the average loss (last line of the table) of MP∪SP is very small, only 0.165%, whereas the loss of the other two variants is more significant (2.237% and 2.977%, respectively). It is interesting to see that the gap varies significantly among the different instances. For a complementary view, looking at Fig. 5 it is possible to notice that the distribution of the outcomes is skewed toward the worse values for the variants of the algorithm that do not use the SP neighborhood (i.e., the two topmost boxplots in the figure), suggesting that this neighborhood is crucial for the solution method. Nevertheless, the interaction of SP with the IRS neighborhood also plays a role in the overall performance of the algorithm allowing for further slight cost improvements. A deeper investigation of the correlation between the features of the instances and the behavior of the neighborhood will be the subject of future work.

In order to have a further insight on the behavior of our search method, in our experiments we tracked the iteration in which the best solution was found. It turned out that in this regard, the behavior is quite irregular, lacking any specific pattern. The average ratio between the iteration of the best and the number of iterations is 0.648, meaning that on average the last 35.2% of iterations are actually “wasted” without improvements. On the other hand, the minimum and the maximum of this ratio are 0.137 and 0.999, respectively, showing that there are both cases in which the waste is much higher and cases with no waste at all in which the method improves until the very end of the search.

Table 12
Ablation analysis: average values of the gaps of the algorithmic variants in comparison to the full neighborhood MP∪SP∪IRS

Instance	MP∪SP	MP∪IRS	MP
0	0.598%	0.448%	0.966%
1	0.031%	3.549%	1.748%
2	−0.885%	−0.011%	0.266%
3	0.181%	0.154%	0.063%
4	0.841%	1.208%	2.216%
5	0.567%	0.375%	1.564%
6	−0.875%	−0.099%	1.034%
7	0.180%	0.135%	0.709%
8	0.612%	0.728%	1.444%
9	1.033%	0.593%	−0.033%
10	0.217%	6.634%	12.865%
11	−0.556%	0.127%	0.478%
12	0.266%	0.525%	0.155%
13	0.450%	0.991%	0.305%
14	1.118%	4.314%	5.742%
15	0.698%	8.541%	12.142%
16	0.467%	2.139%	3.461%
17	0.701%	0.219%	0.394%
18	1.405%	9.211%	9.680%
19	−0.541%	−0.173%	0.384%
20	0.254%	2.490%	1.182%
21	−0.774%	6.475%	8.096%
22	−2.422%	6.386%	8.751%
23	−0.147%	−0.265%	−0.206%
24	1.009%	3.082%	5.222%
25	−0.089%	0.082%	0.048%
26	−0.414%	0.895%	0.674%
27	−0.043%	6.228%	6.681%
28	0.655%	0.887%	2.095%
29	0.399%	1.245%	1.181%
Average	0.165%	2.237%	2.977%

Finally, we discuss the trade-off between travel time and tardiness. To this aim, we run some experiments using alternative weights. In particular, we keep the distance weight fixed and multiply the other two by 1 and 10 alternatively.

In Table 13, we show the average results in comparison with the ones with the original weights. In particular, the column (1,1,1) represents the original weights, the column (1,10,1) the ones where the total tardiness is weighted 10 times and the highest tardiness has the original weight, and so on.

Looking at columns (1,10,1) and (1,1,10), we see that, unsurprisingly, the objective component with the increased weight improves its score at the expense of the distance traveled. The other tardiness-related component is also improved, as the two of them are connected. Notice that the tardiness per service for weights (1,1,10) increases by 20.9%, but this is the average of small quantities sensitive to a few large values. For weights (1,10,10), all tardiness indicators decreased as expected.

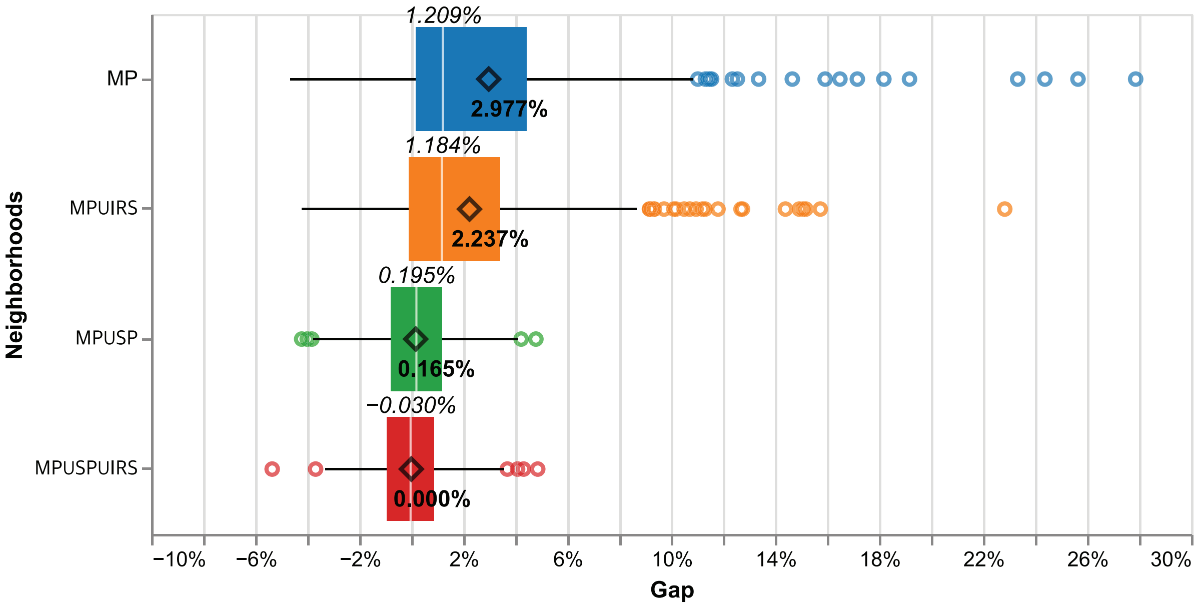

Fig. 5. Distribution of the cost gaps in comparison to the average value obtained by the fully equipped algorithm. The ◇ marker indicates the average gap for that variant of the algorithm. The median values of the distributions are reported in italics.

Table 13
Comparison of new instances with different combinations of the weights.

	(1,1,1)	(1,10,1)		(1,1,10)		(1,10,10)	
	Value	Value	Gap	Value	Gap	Value	Gap
Distance	3143.5	3263.9	3.8%	3189.2	1.5%	3263.6	3.8%
Distance per caregiver	122.8	127.3	3.7%	124.6	1.5%	127.6	3.9%
Total tardiness	30.9	25.7	−16.8%	29.3	−5.0%	21.5	−30.3%
Tardiness per service	0.33	0.32	−3.3%	0.40	20.9%	0.30	−8.8%
Highest tardiness	11.8	11.2	−5.2%	6.7	−43.1%	8.5	−28.4%

However, the differences are pretty small in absolute terms, showing that there is no substantial trade-off between the components. This means that, due to patients' time windows, we cannot eliminate the tardiness by just increasing the traveling time of the caregivers, but we should instead increase their number.

7. Conclusions and future work

In this study, we have introduced an approach based on local search that utilizes a larger neighborhood compared to previous methods proposed in the literature. The efficiency of our method is maintained through the random selection criterion inherent in the SA algorithm, which avoids exhaustive exploration of the entire neighborhood.

Remarkably, our SA approach has yielded favorable comparisons against all state-of-the-art methods in the literature, achieving the best-known results for the majority of instances. While small instances may experience slightly longer running times, these remain within acceptable limits.

An additional contribution of this work is the introduction of a novel benchmark set that extends the existing literature in terms of instance challenges and feature diversity. This new benchmark set not only serves to evaluate the performance of our proposed approach but also provides a more comprehensive and extensive platform for future studies in the field.

In the future, we envision extending the problem formulation by including additional real-world features, such as the synchronization based on mobile equipment, the presence of multiple depots, and penalties for caregiver idleness and overtime. Furthermore, we plan to address the case of the multiday planning horizon, which brings in many new issues related to the stability of caregivers and service time for patients over the days.

These additional considerations will contribute to a more comprehensive and practical formulation of the home healthcare routing and scheduling problem. In addressing this new problem, we plan to extend the current JSON format for input and output data while also translating other datasets, including those by Bredström and Rönnqvist (2008), Grenouilleau et al. (2019), and Di Gaspero and Urli (2014), into the new format.

Acknowledgments

The authors acknowledge the European Commission (Eurostat, Joint Research Centre and DG Regional Policy - REGIO-GIS) for providing the GEOSTAT 1 km^2 population grid and the CINECA consortium for access to the Cloud Computing resources under the Italian Super Computing Resource Allocation Project Grant IA4HC_C.

We wish to thank Alberto Kummer for answering our questions about the work of his research group and for providing us with the CPLEX source code of the MIP model.

This research has been funded by the European Union - *NextGenerationEU*, under the project "Modeling and solving a real-world home healthcare routing and scheduling problem" and by the Italian Ministry of University and Research (MUR) under the PRIN-2020 program under the project "Models and algorithms for the optimization of integrated healthcare management" (no. 2020LNEZYC) supported.

Open access publishing facilitated by Università degli Studi di Udine, as part of the Wiley - CRUI-CARE agreement.

References

Ait Haddadene, S.R., Labadie, N., Prodhon, C., 2016. A GRASP × ILS for the vehicle routing problem with time windows, synchronization and precedence constraints. *Expert Systems with Applications* 66, 274−294.

Bakkehaug, R., Rakke, J.G., Fagerholt, K., Laporte, G., 2016. An adaptive large neighborhood search heuristic for fleet deployment problems with voyage separation requirements. *Transportation Research Part C: Emerging Technologies* 70, 129–141.

Bazirha, M., Benmansour, R., Kadrani, A., 2023a. An efficient two-phase heuristic for the home care routing and scheduling problem. *Computers & Industrial Engineering* 181, 109329.

Bazirha, M., Kadrani, A., Benmansour, R., 2023b. Stochastic home health care routing and scheduling problem with multiple synchronized services. *Annals of Operations Research* 320, 2, 573–601.

Begur, S.V., Miller, D.M., Weaver, J.R., 1997. An integrated spatial DSS for scheduling and routing home-health-care nurses. *Interfaces* 27, 4, 35–48.

Bertels, S., Fahle, T., 2006. A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem. *Computers and Operations Research* 33, 10, 2866–2890.

Birattari, M., Yuan, Z., Balaprakash, P., Stützle, T., 2010. F-race and iterated F-race: an overview. In *Experimental Methods for the Analysis of Optimization Algorithms*. Springer, Berlin, pp. 311–336.

Bredström, D., Rönnqvist, M., 2008. Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. *European Journal of Operational Research* 191, 1, 19–31.

Ceschia, S., Da Ros, F., Di Gaspero, L., Schaerf, A., 2024. Easylocal++ a 25-year perspective on local search frameworks: The evolution of a tool for the design of local search algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 1658–1667. Available at https://bitbucket.org/satt/easylocal-3.

Ceschia, S., Di Gaspero, L., Schaerf, A., 2023. Simulated annealing for the home healthcare routing and scheduling problem. In Dovier, A., Montanari, A., and Orlandini, A. (eds), *AIxIA 2022 – Advances in Artificial Intelligence*, Springer International Publishing, Cham, pp. 402–412.

Cheng, E., Rich, J., 1998. A home health care routing and scheduling problem. Technical Report CAAM TR98–04, Rice University.

Cissé, M., Yalçındağ, S., Kergosien, Y., Şahin, E., Lenté, C., Matta, A., 2017. OR problems related to home health care: a review of relevant routing and scheduling problems. *Operations Research for Health Care* 13–14, 1–22.

Clapper, Y., Berkhout, J., Bekker, R., Moeke, D., 2023. A model-based evolutionary algorithm for home health care scheduling. *Computers and Operations Research* 150, 106081.

Decerle, J., Grunder, O., El Hassani, A.H., Barakat, O., 2018. A memetic algorithm for a home health care routing and scheduling problem. *Operations Research for Health Care* 16, 59–71.

Di Gaspero, L., Urli, T., 2014. A CP/LNS approach for multi-day homecare scheduling problems. In Blesa, M.J., Blum, C., Voß, S. (eds), *Hybrid Metaheuristics*, Springer International Publishing, Berlin, pp. 1–15.

Di Mascolo, M., Martinez, C., Espinouse, M.L., 2021. Routing and scheduling in home health care: a literature survey and bibliometric analysis. *Computers & Industrial Engineering* 158, 107255.

Eveborn, P., Flisberg, P., Rönnqvist, M., 2006. Laps care—an operational system for staff planning of home care. *European Journal of Operational Research* 171, 3, 962–976.

Fedtke, S., Boysen, N., 2017. Gantry crane and shuttle car scheduling in modern rail–rail transshipment yards. *OR Spectrum* 39, 2, 473–503.

Fikar, C., Hirsch, P., 2017. Home health care routing and scheduling: a review. *Computers and Operations Research* 77, 86–95.

Franzin, A., Stützle, T., 2019. Revisiting simulated annealing: a component-based analysis. *Computers and Operations Research* 104, 191–206.

Genet, N., Boerma, W., Kroneman, M., Hutchinson, A., Saltman, R.B., 2012. *Home care across Europe: current structure and future challenges*. World Health Organization, Regional Office for Europe, Copenhagen, Denmark.

Grenouilleau, F., Legrain, A., Lahrichi, N., Rousseau, L.M., 2019. A set partitioning heuristic for the home health care routing and scheduling problem. *European Journal of Operational Research* 275, 1, 295–303.

Grieco, L., Utley, M., Crowe, S., 2021. Operational research applied to decisions in home health care: a systematic literature review. *Journal of the Operational Research Society* 72, 9, 1960–1991.

Grimault, A., Bostel, N., Lehuédé, F., 2017. An adaptive large neighborhood search for the full truckload pickup and delivery problem with resource synchronization. *Computers & Operations Research* 88, 1–14.

Hammersley, J.M., Handscomb, D.C., 1964. *Monte Carlo Methods*. Chapman and Hall, London.

Hiermann, G., Prandtstetter, M., Rendl, A., Puchinger, J., Raidl, G.R., 2015. Metaheuristics for solving a multimodal home-healthcare scheduling problem. *Central European Journal of Operations Research* 23, 89–113.

Kirkpatrick, S., Gelatt, D., Vecchi, M., 1983. Optimization by simulated annealing. *Science* 220, 671–680.

Kordi, G., Divsalar, A., Emami, S., 2023. Multi-objective home health care routing: a variable neighborhood search method. *Optimization Letters* 17, 2257–2298.

Kummer, A.F., 2021. A study on the home care routing and scheduling problem. Ph.D. thesis, Universidade Federal do Rio Grande do Sul.

Kummer, A.F., de Araújo, O.C.B., Buriol, L.S., Resende, M.G.C., 2024. A biased random-key genetic algorithm for the home health care problem. *International Transactions in Operational Research* 31, 3, 1859–1889.

Kummer, A.F., Buriol, L.S., de Araújo, O.C.B., 2020. A biased random key genetic algorithm applied to the VRPTW with skill requirements and synchronization constraints. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, Association for Computing Machinery, New York, NY, USA, pp. 717–724.

Lasfargeas, S., Gagné, C., Sioud, A., 2019. *Solving the Home Health Care Problem with Temporal Precedence and Synchronization*, Springer International Publishing, Cham. pp. 251–267.

Liu, R., Yuan, B., Jiang, Z., 2017. Mathematical model and exact algorithm for the home care worker scheduling and routing problem with lunch break requirements. *International Journal of Production Research* 55, 2, 558–575.

Liu, R., Yuan, B., Jiang, Z., 2019. A branch-and-price algorithm for the home-caregiver scheduling and routing problem with stochastic travel and service times. *Flexible Services and Manufacturing Journal* 31, 989–1011.

Liu, W., Dridi, M., Fei, H., El Hassani, A.H., 2021. Hybrid metaheuristics for solving a home health care routing and scheduling problem with time windows, synchronized visits and lunch breaks. *Expert Systems with Applications* 183, 115307.

Mankowska, D.S., Meisel, F., Bierwirth, C., 2014. The home health care routing and scheduling problem with interdependent services. *Health Care Management Science* 17, 1, 15–30.

Masmoudi, M., Jarboui, B., Borchani, R., 2023. Efficient metaheuristics for the home (health)-care routing and scheduling problem with time windows and synchronized visits. *Optimization Letters* 17, 9, 2135–2167.

Oladzad-Abbasabady, N., Tavakkoli-Moghaddam, R., Mohammadi, M., Vahedi-Nouri, B., 2023. A bi-objective home care routing and scheduling problem considering patient preference and soft temporal dependency constraints. *Engineering Applications of Artificial Intelligence* 119, 105829.

Parragh, S.N., Doerner, K.F., 2018. Solving routing problems with pairwise synchronization constraints. *Central European Journal of Operations Research* 26, 443–464.

Rasmussen, M.S., Justesen, T., Dohn, A., Larsen, J., 2012. The home care crew scheduling problem: preference-based visit clustering and temporal dependencies. *European Journal of Operational Research* 219, 3, 598–610.

Rendl, A., Prandtstetter, M., Hiermann, G., Puchinger, J., Raidl, G., 2012. Hybrid heuristics for multimodal homecare scheduling. In Beldiceanu, N., Jussien, N., Pinson, É. (eds), *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, Springer, Berlin, pp. 339–355.

Soares, R., Marques, A., Amorim, P., Parragh, S.N., 2024. Synchronisation in vehicle routing: classification schema, modelling framework and literature review. *European Journal of Operational Research* 313, 3, 817–840.

Urli, T., 2013. json2run: a tool for experiment design & analysis. *CoRR* abs/1305.1112.

Xiang, T., Li, Y., Szeto, W.Y., 2021. The daily routing and scheduling problem of home health care: based on costs and participants' preference satisfaction. *International Transactions in Operational Research* 30, 1, 39–69.