EURO Journal on Computational Optimization 13 (2025) 100101

Contents lists available at ScienceDirect

EURO Journal on Computational Optimization

journal homepage: www.elsevier.com/locate/ejco

A compact model for the home healthcare routing and scheduling problem

Roberto Montemanni [a,*], Sara Ceschia [b], Andrea Schaerf [b]

[a] *Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Via Amendola, 2, I-42122 Reggio Emilia, Italy*
[b] *Polytechnic Department of Engineering and Architecture, University of Udine, Via delle Scienze 206, I-33100 Udine, Italy*

ARTICLE INFO

Keywords:
Home healthcare
Constraint programming
Lower bounds
Upper bounds

ABSTRACT

Home healthcare has become more and more central in the last decades, due to the advantages it can bring to both healthcare institutions and patients. Planning activities in this context, however, presents significant challenges related to route planning and mutual synchronization of caregivers. In this paper we propose a new compact model for the combined optimization of scheduling (of the activities) and routing (of the caregivers) characterized by fewer variables and constraints when compared with the models previously available in the literature. The new model is solved by a constraint programming solver and compared experimentally with the exact and metaheuristic approaches available in the literature on the common datasets adopted by the community. The results show that the new model provides improved lower bounds for the vast majority of the instances, while producing at the same time high quality heuristic solutions, comparable to those of tailored metaheuristics, for small/medium size instances.

1. Introduction

Home healthcare (HHC) has become popular in the last decades. It consists in moving supportive and geriatric care to the patients' domicile. The reason of this shift is twofold: it offers patients the possibility of staying in a habitual environment and therefore it increases their quality of life. On the other hand, having patients at home decreases the overall healthcare costs substantially [19]. The idea behind the system can be summarized as follows. Treatments are carried out by trained caregivers. A caregiver visits the patient at home within a predefined time window, performs the service operations planned (e.g., medical care or just instrumental activities of daily living), and then travels to a next patient incurring a relevant travel time while moving. The decisions required in the classic activity scheduling problems in hospitals or other healthcare institutions now have a different routing dimension, that makes the problem more challenging. Synchronization issues are also common while planning HHC operations: some medical care activities (e.g., physiotherapy) require the simultaneous presence of more than a single caregiver (e.g., for taking the patient out of the bed). Other activities such as medicine administration or lunch services might require subsequent activities (e.g., a second medicine after a given time or collecting empty lunch boxes) to take place after a certain amount of time.

The motivation of the paper is to advance the quality of mathematical programming models for HHC, and to experimentally show that the new model we propose is able to improve the best-known bounds for different sets of benchmarks commonly adopted in the literature. The contributions of this work can therefore be summarized as follows. First, a new compact model with fewer variables and constraints with respect to those available in the literature, is proposed. Second, the model is expressed according to the syntax

* Corresponding author.
E-mail addresses: roberto.montemanni@unimore.it (R. Montemanni), sara.ceschia@uniud.it (S. Ceschia), andrea.schaerf@uniud.it (A. Schaerf).

https://doi.org/10.1016/j.ejco.2024.100101

Available online 5 December 2024

of the Constraint Programming (CP) solver CP-SAT [44] and tested on three different datasets available from the literature and representing realistic operational cases. New best-known upper and lower bounds for the optimal solution cost of several instances are provided, showing that the new approach is able to provide high quality solutions for instances of sizes normally encountered in real applications (normally up to 100/150 patients). Third, we show how the best metaheuristic methods previously appeared for the problem can potentially exclude from the search space some feasible solutions due to their internal representations.

The rest of the paper is organized as follows: Section 2 provides some references for the aspects of HHC covered by the present work, which is consequently positioned in the available literature. Section 3 presents a formal definition of the problem under investigation and introduces the notation used along the paper. The new model we propose is detailed in Section 4, where some modelling and implementation issues are also discussed. An experimental analysis of the potentials of the new model is provided in Section 5, where the results achieved are compared to those of the state-of-the-art methods from the literature. Conclusions are finally drawn in Section 6.

2. Literature review

The idea of providing healthcare treatments at home has become popular in the last decades, and with it many scientific papers dealing with different aspects of the problem have been published. We focus on those of interested for the present work, treating rostering, scheduling and routing aspects in a deterministic setting. We refer the interested reader to the works by Fikar and Hirsch [18], Cissé et al. [10] and Di Mascolo et al. [16] for a more general literature review on operational research methods for home healthcare routing and scheduling problems. In the remainder of the section we will focus on the works relevant for the setting we consider.

Begur et al. [4] first took into consideration the problem and employed a simple scheduling heuristic to solve the HHC problem. The first Mixed Integer Linear Programming (MILP) model can be traced back to Cheng and Rich [9]. Other MILPs covering a variation of the problem were discussed by Nguyen and Montemanni [40]. A working planning system was presented by Eveborn et al. [17], while temporal dependencies among activities were first considered in the work of Rasmussen et al. [45]. Along the same line, Bredström and Rönnqvist [6] provided a model for the combined routing and scheduling problem with temporal precedence and synchronization constraints, without making explicit reference to home healthcare applications. Models considering the use of public transportation for caregivers (therefore oriented to metropolitan areas), either in a single- or multi-modal fashion, were considered by Bertels and Fahle [5] and Rendl et al. [46]. Temporal horizon spanning on multiple days and a balancing of the workload among caregivers were considered by Di Gaspero and Urli [15]. In addition, they also took into account the possibility not to serve some patients, as often occurs in real-world situations when external caregivers are considered. Their problem was modelled in constraint programming and solved via a Large Neighborhood Search approach.

Population based approaches were tried by a number of authors, in different problem settings. Specifically, Decerle et al. [12] and Grenouilleau et al. [21] devised Memetic Algorithms (i.e., genetic algorithms encapsulating a local search) for the problem, while Clapper et al. [11] presented a model-based Evolutionary Algorithm. In the work by Xiang et al. [49] the problem was formulated as a multi-objective problem trying to balance the total operating cost of caregivers and the patients satisfaction, and solved the problem through a NSGA-II genetic algorithm. A multi-objective perspective was also adopted by Kordi et al. [26] that considered four different objectives (total cost, environmental emission, workload balance, and service quality) and developed a Variable Neighborhood Search method. A recent work that designs and compares many metaheuristic approaches on the problem formulation proposed by Bredström and Rönnqvist [6] is carried out by Masmoudi et al. [36].

The variation of the problem object of the present study, was originally proposed by Mankowska et al. [35] along with a benchmark dataset. The rest of this review focuses on the works dealing with this variation of the problem. As a solution approach, Mankowska et al. [35] described a MILP and developed an Adaptive Variable Neighborhood Search method where the search space is represented by a vector containing the global ordering of patients. The different neighborhood relations reposition a patient in the global order, change the caregiver(s), or swap either the position or the caregiver(s). Lasfargeas et al. [31] developed a two-stage solution approach including a construction heuristic to generate an initial feasible solution, and a Variable Neighborhood Search procedure to explore locally the search space. The method was originally designed for a problem characterized by a horizon spanning multiple days, but results were reported also on the single-day dataset proposed by Mankowska et al. [35]. A Biased Random Key Genetic Algorithm for solving this version of the problem was proposed in the works by Kummer [28] and Kummer et al. [29,30]. In particular, Kummer et al. [30] explored the search space indirectly through a multi-population multi-parent biased random-key genetic algorithm, with an additional component of implicit path-relinking for intensification of the search. The state-of-the-art in terms of heuristic solutions, with respect of the benchmarks introduced by Mankowska et al. [35] and adopted in this paper, are those obtained using Simulated Annealing (SA) by Ceschia et al. [8] and in the subsequent work [7] by the same authors, whose results are available at the repository [25]. They use the same search space of Mankowska et al. namely the global ordering of the patients, but a larger neighborhood which changes the position of a patient in the global ordering and assigns new caregiver(s) to him/her in one single move. These methods are summarized in Table 1, together with other recent and relevant works on other problems addressing single-period home healthcare routing and scheduling problems with different characteristics. The abbreviations used in Table 1 (and in the rest of the paper) for both characteristics and methods, are summarized in Table 2, with a higher level of detail for the methods referred to in the remainder of the paper. The ✓ symbol means that the specific feature is considered in the formulation proposed in the article. When dealing with constraints, “H” means *hard constraints* while “S” marks *soft constraints*.

Table 1
Objectives, constraints and solution methods for single-period home healthcare routing and scheduling problems.

	Objectives						Constraints							Solution method(s)
Reference	FA	OT	TC	TT	UP	WT	CB	MC	PR	PC	SY	SK	TW	
Mankowska et al. [35]	✓	✓		✓						H	H	H	H	AVNS
Lasfargeas et al. [31]	✓	✓		✓						H	H	H	H	VNS
Kummer et al. [29]	✓	✓		✓						H	H	H	H	BRKGA
Kummer et al. [30]	✓	✓		✓						H	H	H	H	BRKGA
Kummer [28]	✓	✓		✓						H	H	H	H	BRKGA-MP-IPR
Ceschia et al. [8]	✓	✓		✓						H	H	H	H	SA
Ceschia et al. [7]	✓	✓		✓						H	H	H	H	MN-SA
Hiermann et al. [23]		✓		✓		✓		H	S			S	S	MA, SA, SS, VNS
Ait Haddadene et al. [1]				✓					S	H	H	H	H	GRASP+ILS
Liu et al. [32]			✓		✓		H		H				H	B&P
Decerle et al. [12]				✓				H			S	H	S	MA
Parragh et al. [43]			✓		✓			H			H	H	H	ALNS
Liu et al. [33]			✓		✓			H				H	S	B&P
Liu et al. [34]			✓				H	H			H	H	H	GA, VNS, SA
Xiang et al. [49]			✓						S			H	H	NSGA
Bazirha et al. [3]		✓	✓								H	H	H	GA, VNS
Bazirha et al. [2]	✓					✓					H	H	H	SA
Clapper et al. [11]		✓	✓			✓			S			H	H	EA
Kordi et al. [26]	✓		✓						S			H	H	VNS
Oladzad-Abbasabady et al. [42]		✓	✓		✓	✓		H	S	S	S	H	S	ILS

Table 2
Abbreviations of objectives, constraints and solution methods for the home healthcare routing and scheduling problems reviewed in Table 1.

Abbreviation	Description
Objective	
FA	Fairness (work balance)
OT	Overtime
TC	Travel cost
TT	Travel time
UP	Unvisited patients
WT	Waiting time
Constraint	
CB	Caregiver break
MC	Multiple centers
PR	Preference
PC	Precedence
SY	Synchronization
SK	Skill requirements
TW	Time windows
WR	Working time regulations
Solution method	
ALNS	Adaptive Large Neighborhood Search
AVNS	Adaptive Variable Neighborhood Search
B&P	Branch & Price
BRKGA	Biased Random-Key Genetic Algorithms
BRKGA-MP-IPR	Biased Random-Key Genetic Algorithms with multi-parents and implicit path-relinking
EA	Evolutionary Algorithm
GA	Genetic Algorithm
GRASP	Greedy Randomized Adaptive Search Procedure
ILS	Iterated Local Search
LNS	Large Neighborhood Search
MH	Metaheuristics
MN-SA	Multi-Neighborhood Simulated Annealing
NSGA	Nondominated Sorting Genetic Algorithm
SA	Simulated Annealing
SS	Scatter Search
VNS	Variable Neighborhood Search

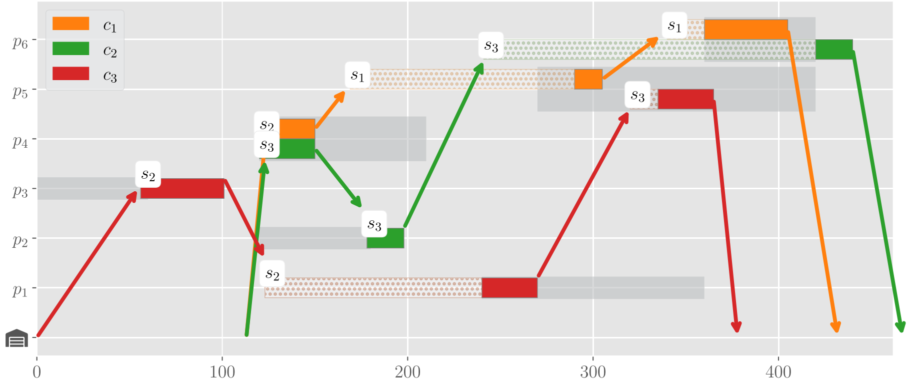

Fig. 1. Optimal solution for the toy-size instance from [25].

3. Formal problem description and notation

The HHC problem under investigation can be described as follows. A set of patients P, a set of service types S, and a set of staff members (caregivers) C are given. Each staff member $k \in C$ is able to perform some of the services, and a binary parameter a_{ks} is equal to 1 if $k \in C$ is qualified to perform the service $s \in S$, and 0 otherwise. The employees start their routes at the HHC's central office, which refers to a given node 0.

The set of patients P is formed by patients of two different types: those requiring a single service, that are part of set P^1, and patients requiring two services, to be provided by two distinct members of the staff, that are part of P^2. Notice that $P = P^1 \cup P^2$.

A binary parameter r_{is} is defined such that $r_{is} = 1$ if patient i requires service $s \in S$, 0 otherwise. We define $S_i \subseteq S$ containing the services requested by patient $i \in P$ (i.e. $r_{is} = 1$). Note that $1 \leq |S_i| \leq 2$ for $i \in P$, while for technical reasons we set $S_0 = S$. We also define $C_i \subseteq C$ as the set of caregivers that have skills to provide at least one service to patient $i \in P$: Caregiver k is in C_i if $\exists s \in S_i$ such that $a_{ks} = 1$. We finally define $P_k \subseteq P$ as the set of patients that can be treated by caregiver $k \in C$ and the central office: $P_k = \{i \in P : k \in S_i\} \cup \{0\}$.

For each double service patient $i \in P^2$ we also have a minimal time distance δ_i^{min} and a maximum time distance $\delta_i^{max} \geq \delta_i^{min}$ that have to be fulfilled between the first and the second service: the second service operation must start at a time within the interval $[t + \delta_i^{min}, t + \delta_i^{max}]$, where t is the starting time of the first service. A precedence relation ($\prec$) exists between the two services required by a patient $i \in P^2$, but when $\delta_i^{min} = \delta_i^{max} = 0$ the service operations have to start simultaneously. It is also assumed that the two operations of a double service have to be executed by different caregivers.

Given the locations of two patients $i, j \in P \cup \{0\}$, the traveling time between them is denoted as τ_{ij}. Given a patient $i \in P$ and a service $s \in S$ such that $r_{is} = 1$, a service time σ_{is} required to perform the treatment is also provided. For each patient $i \in P$, a time window $[e_i, l_i]$, referring to the start of service operations, is also given. If a caregiver arrives at patient i before time e_i, he/she has to wait until the time window opens, otherwise the service is started immediately. Late arrivals of a caregiver (after time l_i) are allowed for the execution of a treatment, but a penalty, proportional to the delay itself, is incurred in such a case.

The objective function to be minimized is the weighted sum of three components. The first component represents the total distance travelled by the caregivers and is a measure of the efficiency of the solution from the viewpoint of the company. The second component accounts for the total tardiness in treatment accumulated by all the caregivers, and can be seen as a measure of the quality of the solution from the viewpoint of the patients. The third component is the maximum tardiness in treatment incurred in the solution, and its role is to avoid all the delay to be concentrated to one (or a few) patients, and therefore should guarantee a fair solution. The three weights, called λ_1, λ_2, and λ_3, are user-defined and model the importance of the different components.

The optimal solution for the toy-size instance of the repository [25] is provided in Fig. 1, where time is represented on the x axis and the depot together with patients $p_1, p_2, \ldots, p_6$ are on the y axis, with time windows depicted as gray blocks. The missions of caregivers c_1, c_2 and c_3 are drawn, with dotted blocks representing waiting times, and full blocks service times (each service executed is identified by a label). Notice that patients p_5 and p_6 require two consecutive services, while patient p_4 needs two caregivers at the same time.

4. A compact mixed integer linear programming model

In this section we describe a compact MILP model for the HHC problem. The model discussed by Mankowska et al. [35], and later revisited by Kummer et al. [30], is equally a MILP, and accounts for $O(|P|^2|C||S|)$ variables and $O(|P|^2|C|^2|S|^2)$ constraints. In this paper, we propose a new model where the size is reduced to $O(|P|^2|C|)$ for both the variables and the constraints. We decided to formalize our model in terms of constraints programming, but the model can be expressed in terms of mixed integer linear programming with no increase in the number of variables or constraints.

By observing that in the model we have no benefit associated with the concept of patients requiring two visits, a simplification can be derived by doubling the nodes of the set P^2, those associated to patients requiring two services. Formally, we define $P' = P \cup P^3$

with $P^3 = \{p'_i : p_i \in P^2\}$, and by assigning a single service to each node of P^2 and P^3 and by imposing a precedence $p_i \prec p'_i$ between nodes (originally the precedence was between the services required by the same node). Finally, after having doubled the patients of the set P^2, each patient $i \in P'$ has now exactly one treatment request, and therefore a univocal service time, referred to as σ_i. The problem can then be modelled as a Vehicle Routing Problem with Soft Time Windows characterized by an articulated objective function and side constraints about the selection of service providers for each node, hard temporal constraints about the visit time of pairs of nodes associated with the same patient and some other details (a formal model will be defined in the remainder of the section). The sets defining the problem are redefined consequently as $P'_k = \{i \in P' : k \in S_i\} \cup \{0\}$, where the definition of each S_i is extended to the new set of nodes P'.

The variables of the corresponding model can be defined as follows:

- x^k_{ij} is a binary variable equal to 1 if caregiver $k \in C$ visits patient $j \in P'_k$ right after patient $i \in P'_k$, 0 otherwise. Furthermore, we also set $x^k_{ii} = 1$ when patient i is *not* visited by the caregiver k; 0 otherwise. This allows to simplify the logic of the constraints of the model.
- t_i is the starting time of the visit at patient $i \in P'$.
- z_i is the delay in the execution of treatment at patient $i \in P'$ with respect to the end of its time window l_i.
- D_{max} is a technical non-negative variable used to capture the maximum tardiness incurred in visiting patients.

The resulting model is as follows.

$$\min \lambda_1 \sum_{i \in P'_0} \sum_{\substack{j \in P' \cup \{0\},\\ j \neq i}} \tau_{ij} \sum_{k \in C_i \cap C_j} x^k_{ij} + \lambda_2 \sum_{i \in P'} z_i + \lambda_3 D_{max} \tag{1}$$

$$s.t.\ D_{max} \geq z_i \qquad i \in P' \tag{2}$$

$$\sum_{j \in P'_k} x^k_{ji} = 1 \qquad k \in C, i \in P_k \tag{3}$$

$$\sum_{j \in P'_k, j \neq i} x^k_{ji} = \sum_{j \in P'_k, j \neq i} x^k_{ij} \qquad k \in C, i \in P'_k \tag{4}$$

$$z_i \geq t_i - l_i \qquad i \in P' \tag{5}$$

$$t_j \geq t_i + \sigma_i + \tau_{ij} - M + M x^k_{ij} \qquad k \in C, i, j \in P'_k, i \neq j \tag{6}$$

$$\delta^{min}_i \leq t_j - t_i \leq \delta^{max}_i \qquad i \in P^2, j \in P^3 : i \prec j \tag{7}$$

$$x^k_{ij} \leq 1 - x^k_{00} \qquad k \in C, i, j \in P'_k, i \neq j \tag{8}$$

$$\sum_{l \in P'_k \setminus \{i\}} x^k_{li} + \sum_{l \in P'_k \setminus \{j\}} x^k_{lj} \leq 1 \qquad k \in C, i, j \in P'_k : i \prec j \tag{9}$$

$$x^k_{ij} \in \{0, 1\} \qquad k \in C, i, j \in P'_k, i \neq j \tag{10}$$

$$t_i \geq e_i \qquad i \in P' \tag{11}$$

$$z_i \geq 0 \qquad i \in P' \tag{12}$$

$$D_{max} \geq 0 \tag{13}$$

The objective function (1) is composed of the three separated components, combined together by the three factors λ_1, λ_2 and λ_3. Constraint (2) captures the maximum among all the z_i values. Constraints (3) impose that, given a caregiver $k \in C$ and a patient $i \in P'$ that can be served by the caregiver, either the patient is visited by the caregiver (one incoming x variable is active for the pair (k, i)) or the variable x^k_{ii} takes value 1. Constraints (4) are flow conservation constraints on the tours of the caregivers, and – when combined with variables t – allow only feasible tours. Inequalities (5) set the value of z variables measuring delays based on the values of variables t modelling visiting times and latest times of time windows l_i. Constraints (6) regulate timings and are based on a sufficiently large constant M. Each constraint becomes active only when caregiver k visits patient j right after patient i, and regards the reciprocal timing of the two services provided. Inequalities (7) ensure that patients requiring two treatments receive them in the proper (hard) time separation. Constraints (8) ensure that no operation is carried out by a caregiver not leaving the central office. Constraints (9) impose that patients requiring two services are taken care by two distinct caregivers (as explicitly requested by the problem). Finally, constraints (10)–(13) define the domain of the variables. Note that t variables cannot be assigned to values lower than the earlier arrival times of the respective patients.

4.1. Optimizing the model for the CP-SAT solver

Recently, models with characteristics similar to those of the model we propose (see, for example, Montemanni and Dell'Amico [37], Montemanni et al. [39], Dell'Amico et al. [13] and [14]) have been successfully attacked by the CP-SAT solver [44], part of the OR-Tools suite [20] by Google, that has proven superior to traditional MILP solvers such as Gurobi [22] or CPLEX [24] on these specific models. For this reason, we plan to use such a solver and here we describe the modification made to the model to fully exploit the CP-SAT solver.

The CP-SAT solver is base on the efficient use of a multi-threaded environment, compatible therefore with modern processors. The solver operates similarly to a portfolio strategy, where diverse methods work concurrently and exchanging limited data among threads. The main process runs a Constraint Programming solver based on a Lazy Clause Generation (LCG, Stuckey [47]), but other unrelated approaches work in parallel to support the main thread and exchange information such as new bounds and solutions. The idea behind LCG involves the transformation of the whole problem into a SAT-formula, subsequently employing a SAT-solver to look for solutions (or prove bounds by infeasibility). A (partial) linearization of the model is also created, and the corresponding linear program gets solved with simplex algorithms, while other classic MILP techniques are run to enhance bounds and retrieve new solutions, aiming at supporting the satisfiability model. Different instances of a Large Neighborhood Search (LNS) metaheuristic, seeking for high-quality feasible solutions, are finally executed, aiming to unearth high-quality feasible solutions. While the general idea might appear inefficient due to potential redundancy, this approach demonstrates effectiveness in practice. The rationale behind this lies in the inherent challenge of predicting which algorithm is best suited to solve a given problem (No Free Lunch Theorem, Wolpert and Macready [48]). Consequently, a pragmatic strategy involves running diverse approaches simultaneously, with the hope that at least one will effectively address the problem at hand. In contrast, branch and cut-based MILP solvers such as Gurobi [22] or CPLEX [24] execute a more streamlined partitioning of the search space, thereby minimizing redundancy. However, they adhere to a specific strategy, which might not always align with the optimal choice for every scenario, although it might be extremely efficient for others.

Going back to efficient representations of the new model in terms of CP-SAT, inequalities (2) can be expressed through the following compact constraint, using the *AddMaxEquality* statement of CP-SAT:

$$D_{max} = \max(z_i, \text{ with } i \in P') \tag{14}$$

Inequalities (6) can be changed to the following set of constraints, that use channeling through the *OnlyEnforceIf* statement of CP-SAT, here indicated with the notation "⇒":

$$x_{ij}^k \Rightarrow t_j \geq t_i + \sigma_i + \tau_{ij} \quad k \in C, i, j \in P'_k, i \neq j \tag{15}$$

Inequalities (8) can be changed to the following constraints, making use of constraint channeling and of the negation operator, here indicated as "¬", and implemented via the *Not* statement of CP-SAT to negate a Boolean variable:

$$x_{00}^k \Rightarrow \neg x_{ij}^k \quad k \in C, i, j \in P'_k, i \neq j \tag{16}$$

The global constraint *AddBoolOr* of CP-SAT, operating over Boolean variables and here indicated as "∨", can be used to represent inequalities (9).

$$x_{ii}^k \vee x_{jj}^k \quad k \in C, i, j \in P'_k : i \prec j \tag{17}$$

Finally, in order to handle non-integer input data into CP-SAT (that only deals with integers), it is necessary to scale up all the interested values by a given factor F, and then scale down the results obtained consequently. In our implementation we set $F = 10000$, as recommended by Montemanni and Dell'Amico [38] for a similar application.

These settings will be kept for all the experiments with CP-SAT reported in this paper.

Some experimental results comparing CP-SAT with the state-of-the-art MILP solver Gurobi can be found in Appendix A.

5. Computational experiments

In this section the constraint programming model discussed in Section 4 is evaluated from an experimental perspective, and positioned with respect to the state-of-the-art approaches currently available.

The model described in Section 4 was implemented in Python and solved via the CP-SAT solver of Google OR-tools (v. 9.7) [44] using standard settings. Whenever available, for lower bounds on the cost of the optimal solution (LB) we consider that obtained in Kummer et al. [30] by solving a Mixed Integer Linear Program originally proposed in Mankowska et al. [35] (MILP), or the linear relaxation of the same model (LR) from Kummer [28]. For the heuristic solutions (UB) we compare with those obtained by Mankowska et al. [35] and Lasfargeas et al. [31] using Variable Neighborhood Search (VNS), Kummer [28], Kummer et al. [30] and Kummer et al. [29] by using Biased Random-Key Genetic Algorithms (BRKGA), Kummer [28] using an evolution of the same algorithm involving multi-parents and implicit path-relinking (BRKGA-MP-IPR), Ceschia et al. [8] using Simulated Annealing (SA) and those published in the repository [25] and obtained with a Multi-Neighborhood version of the same algorithm (MN-SA). Each new best lower or upper bound retrieved by the new model is highlighted in bold, while each suboptimal bound is in italics.

We consider in our experimental analysis three datasets: the one from Mankowska et al. (available at [41]), the one by Kummer (available at [27]), and the one by the IOLab (available at [25]). The datasets have different original data formats, but all three can be retrieved from [25] in a single, novel format, based on JSON. The remainder of this section is organized based on the dataset analyzed.

Table 3
Results on the small instances from Mankowska et al. [35].

Instances	MILP[a]	VNS[b]		VNS[c]		BRKGA[d]		BRKGA[e]		SA[f]		MN-SA[g]	CP-SAT[h]	
	LB	UB	Sec	UB	Sec	UB	Sec	UB	Sec	UB	Sec	UB	LB	UB
B1	428.1	458.9	< 1	434.1	53.1	428.1	8.6	428.1	0.8	428.1	70.2	428.1	428.1	428.1
B2	476.0	476.2	< 1	476.0	27.7	483.6	8.4	476.0	0.9	476.0	68.9	476.0	476.0	476.0
B3	399.1	399.2	< 1	399.1	63.5	402.8	9.0	402.8	1.0	399.1	70.7	399.1	399.1	399.1
B4	411.3	576.0	< 1	414.0	66.8	420.3	8.2	422.1	1.1	411.3	68.7	411.3	411.3	411.3
B5	366.3	391.1	< 1	385.6	13.7	372.2	8.2	369.4	1.0	366.3	68.8	366.3	366.3	366.3
B6	405.6	534.7	< 1	447.8	443.7	471.0	8.9	470.6	1.2	464.6	70.1	464.6	*393.1*	**440.9**
B7	328.7	355.5	< 1	328.7	61.5	328.7	9.5	328.7	0.9	328.7	68.6	328.7	328.7	328.7
B8	357.7	357.8	< 1	359.7	79.3	359.7	9.2	357.7	0.7	357.7	70.6	357.7	357.7	357.7
B9	330.3	403.8	< 1	404.1	62.1	402.7	10.0	404.1	0.9	402.7	71.9	402.7	*319.3*	402.7
B10	421.0	500.4	< 1	462.7	8.7	469.6	9.2	469.6	0.9	462.8	69.8	462.7	*414.2*	462.7
Avg	392.4	445.4	< 1	411.2	88.0	413.9	8.9	412.9	0.9	409.7	69.8	409.7	*389.4*	**407.4**
C1	459.3	1123.6	< 1	974.2	96.2	965.2	36.9	969.1	3.1	943.7	96.0	943.7	**499.0**	943.7
C2	373.9	673.8	< 1	605.1	106.4	583.4	39.0	584.2	2.9	569.1	93.9	569.4	**400.4**	569.1
C3	390.5	642.4	< 1	562.9	109.8	548.8	37.4	549.6	2.9	537.8	94.7	541.1	**419.8**	537.8
C4	372.0	580.4	< 1	521.9	112.4	519.9	36.2	520.1	3.0	495.2	94.0	495.2	**417.9**	495.2
C5	465.0	754.6	< 1	683.1	114.9	678.6	31.4	668.7	3.4	655.7	93.2	655.7	**516.6**	655.7
C6	360.7	951.6	< 1	854.6	115.9	840.7	37.2	841.5	2.9	815.2	96.0	813.3	**382.2**	813.3
C7	354.2	577.4	< 1	529.2	109.4	534.9	42.2	533.9	3.4	514.1	92.8	511.9	**385.9**	511.9
C8	375.5	540.6	< 1	471.0	110.8	474.6	36.5	476.0	3.5	469.5	93.7	469.0	**403.3**	*469.5*
C9	355.3	608.7	< 1	551.1	115.4	534.3	42.8	545.2	3.4	533.1	94.5	535.1	**407.7**	**527.7**
C10	431.2	679.3	< 1	608.9	99.0	611.3	35.3	611.0	2.7	590.3	93.5	590.3	**459.4**	590.3
Avg	393.8	713.2	< 1	636.2	109.0	629.1	37.5	629.9	3.1	612.4	94.2	612.5	**429.2**	**611.4**
D1	492.1	1321.8	5.0	1278.2	143.0	1186.2	93.7	1193.2	8.0	1122.8	120.6	1111.4	**503.3**	**1104.3**
D2	384.7	892.7	4.0	746.9	168.7	693.3	82.9	679.6	7.2	649.1	119.0	652.2	**461.5**	*653.3*
D3	380.1	819.4	4.0	678.6	155.4	635.7	102.2	644.2	8.5	616.4	118.8	622.6	**436.1**	**612.9**
D4	418.9	877.4	4.0	809.7	148.5	814.4	82.3	795.2	7.2	776.6	119.5	773.0	**453.5**	**772.4**
D5	415.8	872.1	5.0	777.0	150.3	691.5	92.4	693.8	7.7	656.9	120.2	653.7	**438.4**	**649.7**
D6	392.1	835.2	5.0	768.6	154.6	733.7	105.8	731.7	7.9	688.6	119.3	688.2	**407.9**	*697.6*
D7	372.5	706.3	6.0	600.1	168.1	590.6	112.8	586.1	7.8	566.2	118.7	564.6	**426.6**	*573.1*
D8	409.4	811.4	4.0	715.5	149.8	661.8	102.7	658.5	8.1	650.0	117.4	650.6	**486.5**	*658.2*
D9	385.9	842.7	6.0	741.0	156.0	706.1	92.6	689.8	9.2	651.4	119.2	650.3	**438.1**	*655.2*
D10	485.6	1306.6	3.0	1424.6	173.1	1208.7	77.7	1189.3	6.9	1157.6	120.4	1152.1	**539.0**	*1161.9*
Avg	413.7	928.6	4.6	854.0	156.8	792.2	94.5	786.1	7.9	753.6	119.3	751.9	**459.1**	*753.9*

[a] Kummer et al. [30], CPU Intel Xeon E5-2697v2, 12x2.70 GHz; RAM 64 GB; IBM CPLEX 20.1.0.0 solver; 7200 sec time limit.
[b] Mankowska et al. [35], CPU Intel Core, 3.40 GHz.
[c] Lasfargeas et al. [31], CPU Intel Core i7, 4x4.0 GHz; RAM 32 GB; 40 runs, each with 1800 sec time limit.
[d] Kummer et al. [29], CPU Intel Core i7-3612QM, 4x2.10 GHz; RAM 8 GB; 15 runs.
[e] Kummer et al. [30], CPU Intel Xeon E5-2697v2, 12x2.70 GHz; RAM 64 GB; 20 runs.
[f] Ceschia et al. [8], CPU Intel Core i7-7700, 4x3.60 GHz; 10 runs.
[g] IOLab [25], CPU Intel Core i7-7700, 4x3.60 GHz; mixed computation times; unrecorded number of runs.
[h] CPU Intel Core i7 12700F – 12x2.1 GHz; RAM 32 GB; OR-Tools CP-SAT 9.7 solver; 3600 sec time limit.

5.1. Results on the dataset from Mankowska et al.

A consolidated dataset adopted to benchmark methods for the HHC problem is the one proposed by Mankowska et al. [35], where the interested reader is referred to for full details. The dataset is composed of seven groups – labelled from A to G – of ten instances each, with 10, 25, 50, 75, 100, 200, 300 patients, respectively, for a total of 70 instances. The objective function weights are set as $\lambda_1 = \lambda_2 = \lambda_3 = 1/3$. The number of services provided is always 6, with a number of caregivers ranging from 3 to 40. The number of patients requiring a double visit is between 30% and 33% of the total number of patients. In this work, we only consider groups B to G, ranging from 25 and 300 patients, being group A not challenging enough for any of the methods.

The results are split into Tables 3 and 4. Having this dataset being around for a fairy long time now, results have been reported for many different approaches.

The results suggest that the new model leads to high quality lower bounds, better than the previously best-known ones for almost all the instances considered in the tables, and notwithstanding that the previous best method by Kummer et al. [30] had been run for double the time with respect to our model. Concerning the quality of the heuristic solutions provided by the new method (*UB*), we can see that state-of-the-art or near-state-of-the-art results are reported up to 100 patients (group *E*). This result is remarkable since we compare with purposely-designed metaheuristic methods fully exploiting problem information that cannot be captured by our model. No feasible solution is found for the larger instances, clearly highlighting a threshold after which the new model suffers from scalability issues – although such issues seem to appear later with respect to the other exact solvers – and is not effective as a heuristic solver. It is also important to observe how well the method performs on the medium size instances (Table 3) with a number of patients ranging from 25 to 75: six new best-known heuristic solutions were retrieved (namely for the instances with a bold entry

Table 4
Results on the large instances from Mankowska et al. [35].

Instances	MILP[a]	VNS[b]		BRKGA[c]		BRKGA[d]		SA[e]		MN-SA[f]	CP-SAT[g]	
	LB	UB	Sec	UB	Sec	UB	Sec	UB	Sec	UB	LB	UB
E1	430.4	1604.9	17.0	1331.5	193.6	1327.7	17.2	1260.6	142.7	1255.9	**572.3**	*1275.4*
E2	444.9	1101.9	10.0	848.1	192.0	829.8	17.1	782.1	139.6	778.4	**531.2**	*817.0*
E3	454.3	986.4	14.0	788.0	182.9	789.6	16.7	763.4	142.2	757.8	**506.0**	*760.1*
E4	412.1	871.0	19.0	711.2	196.5	723.9	14.8	691.2	142.2	687.2	**444.2**	*730.3*
E5	416.6	1018.0	19.0	781.5	182.3	780.0	17.0	713.8	142.1	707.7	**453.5**	*768.4*
E6	416.6	1003.0	19.0	790.5	177.6	779.8	18.3	753.6	143.3	748.9	**435.7**	*766.8*
E7	389.6	921.1	20.0	711.1	191.8	705.8	18.0	682.7	141.6	679.0	**451.9**	*691.4*
E8	433.9	884.6	19.0	752.4	168.8	733.9	17.2	710.5	143.5	707.1	**490.0**	*723.9*
E9	446.5	1131.7	18.0	921.8	163.1	893.4	16.4	859.2	141.3	857.4	**545.9**	*859.2*
E10	455.1	1053.6	11.0	825.2	174.5	822.9	16.0	788.7	140.7	788.1	**505.0**	*828.4*
Avg	430.0	1057.6	16.6	846.1	182.3	838.7	16.9	800.6	141.9	796.8	**493.6**	*822.1*
F1	548.9	1721.4	889.0	1402.0	745.5	1311.1	124.4	1246.1	248.2	1237.0	**618.2**	-
F2	543.3	1763.8	909.0	1336.3	812.1	1298.3	121.7	1226.1	246.1	1206.9	**582.2**	-
F3	547.6	1549.6	868.0	1263.4	780.3	1216.0	116.5	1161.9	245.6	1128.6	**559.3**	-
F4	531.8	1420.4	1321.0	1124.2	901.9	1100.7	136.0	1055.2	246.9	1037.0	**547.7**	-
F5	538.1	1701.9	1145.0	1329.3	826.1	1298.6	119.8	1231.6	246.1	1191.5	**561.8**	-
F6	518.5	1639.7	836.0	1332.1	649.8	1292.5	109.6	1237.1	247.9	1219.8	*452.7*	-
F7	513.0	1384.3	1294.0	1131.3	817.0	1084.6	120.5	1073.2	245.5	1063.0	**590.9**	-
F8	536.2	1544.6	924.0	1132.8	716.4	1123.2	107.7	1089.4	246.4	1070.9	**598.6**	-
F9	543.2	1572.9	1642.0	1311.4	770.4	1263.2	125.4	1204.2	246.8	1177.9	**574.1**	-
F10	546.8	1581.0	1326.0	1418.5	740.3	1383.1	119.6	1270.5	248.6	1260.0	*541.7*	-
Avg	536.7	1588.0	1115.4	1278.1	776.0	1237.1	120.1	1179.5	246.8	1159.3	**562.7**	-
G1	612.4	2248.0	7200.0	1778.5	1949.8	1744.1	439.4	1681.1	367.7	1650.2	**663.6**	-
G2	605.8	2316.1	7200.0	1824.7	2115.1	1709.7	519.5	1652.7	362.0	1591.8	**723.1**	-
G3	614.2	1885.3	7147.0	1514.2	1935.1	1464.7	461.6	1432.4	367.4	1379.7	*601.1*	-
G4	604.3	2023.2	7200.0	1564.4	2137.6	1508.9	529.0	1458.5	373.6	1418.6	**701.7**	-
G5	633.7	2247.6	7200.0	1698.3	1840.9	1652.9	466.6	1550.8	371.3	1540.9	**656.9**	-
G6	621.5	2144.4	7200.0	1714.4	2014.4	1681.6	570.5	1654.7	371.5	1619.5	*613.1*	-
G7	602.4	1971.5	6934.0	1640.1	1844.3	1536.0	522.4	1493.9	365.1	1468.5	**723.1**	-
G8	618.7	1987.4	7200.0	1547.6	1799.1	1498.4	531.7	1457.0	367.3	1420.9	**644.0**	-
G9	662.7	2415.5	7023.0	1942.2	1810.7	1850.1	446.6	1768.9	371.3	1742.8	**699.5**	-
G10	633.8	2373.4	7003.0	1872.1	1649.6	1785.4	482.8	1690.6	372.4	1634.0	**690.9**	-
Avg	620.9	2161.2	7130.7	1709.7	1909.7	1643.2	497.0	1584.0	369.0	1546.7	**671.7**	-

[a] Kummer et al. [30], CPU Intel Xeon E5-2697v2, 12x2.70 GHz; RAM 64 GB; IBM CPLEX 20.1.0.0 solver; 7200 sec time limit.
[b] Mankowska et al. [35], CPU Intel Core, 3.40 GHz.
[c] Kummer et al. [29], CPU Intel Core i7-3612QM, 4x2.10 GHz; RAM 8 GB; 15 runs.
[d] Kummer et al. [30], CPU Intel Xeon E5-2697v2, 12x2.70 GHz; RAM 64 GB; 20 runs.
[e] Ceschia et al. [8], CPU Intel Core i7-7700, 4x3.60 GHz; 10 runs.
[f] IOLab [25], CPU Intel Core i7-7700, 4x3.60 GHz; mixed computation times; unrecorded number of runs.
[g] CPU Intel Core i7 12700F – 12x2.1 GHz; RAM 32 GB; OR-Tools CP-SAT 9.7 solver; 3600 sec time limit.

in the last column of Table 3). It should be observed that the CP-SAT solver was run for one hour, while the metaheuristics algorithms have running times that are one order of magnitude shorter. On the other hand, the best result over several runs is reported for them, re-balancing the fairness of the comparison. Concerning lower bounds, the new model is able to improve the best-known solution for all the instances but fourteen, matching anyway the best-known bound for seven of the latters.

5.2. Results on the instances from Kummer

According to the analysis reported by Kummer [28], the instances of the dataset proposed in Mankowska et al. [35] lack a structure in the geographic data, making them less realistic. The author proposes a new instance generator that overcomes this weakness by using data from real cities and computing real routes among randomly selected addresses representing patients and offices. This implies that the resulting instances are neither Euclidean nor symmetric. The ratio of patients requiring double visits is set exactly as in the dataset from Mankowska et al. and so is the number of services, fixed again to 6. The distribution of the skills of the caregivers is however more balanced, in order to avoid the case of services provided by a very small set of caregivers only. The instances are based on geographic data, and rely on three features: the node generation strategy (random or clustered), the central office placement strategy and the cluster density. A total of 22 different combinations of values for these features were selected and 100 instances were generated for each combination, for 8 different sizes (ranging between 10 and 400), thus creating a large set of 17,600 (22 × 8 × 100) instances for training purposes. A selection of 160 instances was extracted as the validation set. Among those, we selected 120 instances because we decided not to report results for the instances with 300 and 400 patients since no feasible solution was retrieved by the new model and just a few lower bounds were improved, all very marginally. The instances are available from [27].

The results are summarized in Tables 5 and 6. Only for Table 5, a new column labelled OPT has been added to the CP-SAT results. When an instance is solved to optimality for the first time, the corresponding row will contain an asterisk. Being the examined dataset more recent, a limited number of methods is available for comparison purposes. The results achieved by solving the new model have a similar trend to that depicted in Section 5.1 for the previous dataset. The new lower bounds in this case are prominently better than those previously reported (they improve the state-of-the-art for all the instances but five, and even for these instances the state-of-the-art lower bound is matched). Concerning heuristic solutions, also for this dataset we were able to retrieve four new best-known solutions (bold entries in the last but one column), while matching or almost-matching all the best-known solutions for the instances up to 50 patients, and being anyway very close to the state-of-the-art for the instances with up to 100 patients. Finally, the new model allowed to close for the first time all the instances with 10 patients and half of those with 25 (last column of Table 5). In conclusion, solving the model we propose remains a viable way for having good solutions without having to devise a specialized metaheuristic method, at least for small/medium problems.

5.3. Results on the instances from IOLab

Another step towards real settings was taken with the new dataset available in the repository [25] from IOLab, where instances taking into account both the real distribution of population in a territory and the actual road distances were proposed. The aim was to establish a new valid and modern benchmark set for the research community. A variety of areas in Italy, including urban, rural, and mountainous regions, which possess varying features with respect to population, morphology, urban sprawl, and compactness, were selected to generate the benchmarks. Within each chosen area, the locations of the patients were sampled according to the population of each subarea, then travel times between each pair of points were calculated by considering the real road network and traffic conditions. A dataset of 200 instances was created by sampling various Italian territories of different size and population density. Differently from the previous two datasets, the values of the main features, such as patients, caregivers, and services, were selected randomly for each instance. Of the 200 instances, 30 form the benchmark dataset, while the remaining 170 are meant for training purposes. Full details about the instances and their features can be found in [25]. The computational results are summarized in Table 7. For this dataset only the upper bounds reported in the repository [25] are available. We here disclose the first-ever lower bounds, obtained by solving the new model. As for the previous datasets, the lower bounds are sharper for small- and medium-size instances, while they become loose with respect to the available upper bounds for the larger ones. In our opinion the heuristic solutions reported are already high-quality, while the lower bounds are perfectible, due to scalability issues encountered in solving the new model. This conjecture is corroborated by the few heuristic solutions retrieved by the new model on these instances, that show once again the scalability issues of the model. The few solutions retrieved are however of acceptable quality, although not matching those reported in the repository [25].

5.4. Considerations on the results

It is worth mentioning that our model is totally general and imposes no additional constraints on the structure of the solution. This is not the case for the approach by Mankowska et al. [35], followed also by Ceschia et al. [8], that is based on a strict global ordering of the patients. The consequence of this global ordering is that it is not possible to reach a solution in which two double-service patients are visited by the same two caregivers, but in reverse order. For example, our solution of instance B_6 from the dataset Mankowska et al. [35], shown in Fig. 2 and represented according to the same logic of Fig. 1, exhibits this behavior for patients p_{22} and p_{25} with caregivers c_1 and c_3 (see top-left part of the figure highlighted with a dashed rectangular box). Indeed, this solution has a cost of 440.9, that is never obtained by any of the cited papers. For example, Ceschia et al. [8] consistently find the best-known value for all the instances of the group B, except for B_6, where it is stuck to the value 464.6. However, it should be observed that the limitation above has a marginal impact on the searching capabilities of the methods, while simplifying a lot their logic. The engineering idea remains therefore extremely valid in our opinion. Notice also that caregiver c_2 is not leaving the central office, thus the route is empty.

A further experiment we carried out is to feed the CP-SAT solver (again running with standard settings) with the best-known heuristic solutions available in the repository [25], and reported in the *MN-SA* columns of the tables previously disclosed. The outcome of this study was that the quality of the lower bounds is consistently and substantially degraded, while some improved heuristic solutions are often found in a few to some hundred seconds. However, the improvements are always very limited, normally consistently below 1%, making the computational effort not fully justified in our opinion. This however clearly indicates that the heuristic methods available are extremely good in identifying the most promising regions of the search space, but there is still some margin of improvement in their capability.

6. Conclusions

Home healthcare has been seen in the last few years both as a solution to reduce the costs of healthcare institutions and a way to have happier patients, since they can stay at home with their families. Such a system poses however new challenges when it is time to plan the activities of the caregivers, since now they not only have to carry out treatments, but also have to move among patients. Solving methods need to schedule the activities of caregivers at patients, which also implies routing the caregivers around, in such a way that all the requirements of the patients are attended.

Table 5
Results on the small instances from Kummer [28].

Instances	LR[a]	BRKGA[b]	BRKGA-MP-IPR[c]		MN-SA[d]	CP-SAT[e]		
	LB	UB	UB	Sec	UB	LB	UB	OPT
10_3_11_0.4_R_C	61.0	110.3	110.3	44.0	106.7	**106.7**	106.7	*
10_3_16_1.6_C_C	31.0	183.7	183.7	44.3	183.7	**183.7**	183.7	*
10_3_22_1.6_R_C	42.0	79.0	79.0	44.1	79.0	**79.0**	79.0	*
10_3_35_1.0_R_R	55.7	385.0	385.0	43.9	385.0	**385.0**	385.0	*
10_3_40_0.8_R_C	18.1	129.7	129.7	43.9	129.7	**129.7**	129.7	*
10_3_40_1.0_R_C	18.1	129.7	129.7	44.0	129.7	**129.7**	129.7	*
10_3_46_1.6_R_C	33.4	86.0	86.0	43.9	86.0	**86.0**	86.0	*
10_3_51_1.0_R_RC	54.7	97.0	97.0	44.1	91.7	**91.7**	91.7	*
10_3_56_1.2_R_RC	42.4	92.3	92.3	51.6	88.0	**88.0**	88.0	*
10_3_60_1.0_R_RC	57.0	83.0	83.0	44.0	83.0	**83.0**	83.0	*
10_3_80_1.0_C_R	42.4	102.3	102.3	61.0	102.3	**102.3**	102.3	*
10_3_81_1.2_R_C	38.1	241.7	241.7	43.8	234.0	**234.0**	234.0	*
10_3_86_1.2_C_C	31.8	198.7	198.7	43.8	198.7	**198.7**	198.7	*
10_3_88_1.0_R_C	17.7	113.7	113.7	43.8	113.7	**113.7**	113.7	*
10_3_88_1.2_R_C	17.7	113.7	113.7	44.1	113.7	**113.7**	113.7	*
10_3_88_1.6_C_C	14.4	89.3	89.3	43.8	87.0	**87.0**	87.0	*
10_3_88_1.6_R_C	16.7	150.7	150.7	43.7	150.7	**150.7**	150.7	*
10_3_90_1.2_C_C	42.3	64.3	64.3	43.8	64.3	**64.3**	64.3	*
10_3_90_1.2_C_RC	49.7	75.0	75.0	43.9	75.0	**75.0**	75.0	*
10_3_96_1.0_R_C	36.8	126.7	126.7	44.2	123.7	**123.7**	123.7	*
Avg	36.0	132.6	132.6	45.2	131.3	**131.3**	131.3	
25_5_21_1.6_R_RC	93.7	205.0	204.7	50.0	197.7	**197.7**	197.7	*
25_5_22_1.0_C_C	55.0	899.3	820.3	49.1	820.3	**217.8**	820.3	
25_5_26_0.8_C_C	62.4	956.3	956.3	49.2	952.3	**246.0**	952.3	
25_5_37_0.8_R_RC	93.4	1441.3	1434.7	49.2	1434.7	**274.8**	1434.7	
25_5_37_1.0_R_RC	93.4	1441.3	1434.7	49.2	1434.7	**270.0**	1434.7	
25_5_42_0.8_C_RC	101.0	287.7	287.7	49.4	287.3	**287.3**	287.3	*
25_5_42_0.8_R_RC	93.1	1196.7	1161.7	49.0	1158.3	**254.1**	*1159.0*	
25_5_47_0.8_R_C	53.7	848.7	848.0	49.2	842.3	**246.0**	*849.0*	
25_5_48_1.0_C_C	78.4	280.3	275.0	49.5	272.0	**272.0**	272.0	*
25_5_49_0.4_C_C	86.3	141.7	138.7	49.9	137.7	**137.7**	137.7	*
25_5_56_0.8_R_RC	63.3	179.7	178.3	50.1	177.3	**177.3**	177.3	*
25_5_69_1.6_R_C	52.7	715.3	695.0	49.1	695.0	**252.9**	695.0	
25_5_69_1.6_R_RC	72.7	1171.0	1149.0	49.0	1146.0	**112.0**	1146.0	
25_5_6_0.8_R_C	48.4	706.0	681.0	49.6	678.7	**223.2**	678.7	
25_5_70_1.0_R_C	65.0	146.0	146.0	50.0	141.7	**141.7**	141.7	*
25_5_73_0.4_R_RC	73.7	155.3	151.0	49.6	150.0	**150.0**	150.0	*
25_5_76_1.2_C_C	71.7	184.0	183.7	49.7	183.7	**183.7**	183.7	*
25_5_80_0.8_C_RC	116.1	437.3	436.7	49.1	436.3	**436.3**	436.3	*
25_5_96_0.8_R_C	57.4	730.3	707.7	49.5	697.7	**176.3**	697.7	
25_5_97_1.0_R_R	123.4	258.7	258.7	49.4	258.7	**258.7**	258.7	*
Avg	77.7	619.1	607.4	49.4	605.1	**225.8**	*605.5*	
50_10_100_1.6_C_C	75.0	162.3	159.0	61.9	154.7	**139.6**	*155.0*	
50_10_26_1.0_C_RC	121.0	629.3	610.3	59.8	607.0	**180.4**	607.0	
50_10_26_1.2_R_C	67.3	349.3	312.0	60.9	304.0	**175.0**	*307.3*	
50_10_2_1.0_C_R	155.7	370.7	345.0	60.8	345.3	**294.4**	**344.7**	
50_10_32_1.2_R_C	85.7	473.7	442.3	60.8	434.3	**167.5**	**430.0**	
50_10_33_0.4_R_RC	140.3	287.0	279.7	62.1	273.3	**222.6**	*279.0*	
50_10_3_0.8_R_C	112.0	617.3	589.0	60.9	585.7	**210.3**	*586.0*	
50_10_3_1.6_C_C	66.7	272.7	262.0	61.6	251.0	**162.1**	*251.7*	
50_10_44_1.6_R_C	60.0	305.3	279.3	62.6	265.7	**82.0**	*267.3*	
50_10_53_1.2_R_C	82.7	261.3	250.7	60.6	246.0	**166.6**	*251.0*	
50_10_60_0.4_C_RC	107.0	281.3	266.3	62.3	261.7	**200.3**	**260.7**	
50_10_61_1.0_R_RC	107.0	288.3	276.3	62.7	270.0	**210.5**	*276.7*	
50_10_62_1.2_R_C	74.0	216.0	211.3	62.8	204.7	**137.1**	204.7	
50_10_74_1.0_R_RC	113.7	319.0	290.3	60.3	289.7	**204.0**	*290.7*	
50_10_80_0.4_C_C	144.7	788.7	762.7	58.8	753.3	**267.3**	*755.0*	
50_10_80_1.0_C_C	102.2	855.3	826.7	58.3	815.7	**241.0**	815.7	
50_10_80_1.0_C_RC	152.0	1081.3	1057.3	58.3	1045.0	**279.5**	*1048.7*	
50_10_81_1.0_R_RC	133.4	1053.7	999.7	58.1	986.3	**270.6**	986.3	
50_10_88_1.6_C_RC	130.0	342.7	317.7	61.9	308.0	**235.9**	*320.7*	
50_10_97_0.8_C_RC	129.7	784.0	759.7	59.2	749.7	**243.0**	*763.0*	
Avg	108.0	487.0	464.9	60.7	457.6	**204.5**	*460.0*	

[a] Kummer [28], CPU Intel i7-930, 2.80 GHz, IBM CPLEX 20.1.0.0 solver; unreported time.
[b] Kummer [28], CPU Intel i7-930, 2.80 GHz; unreported time.
[c] Kummer [28], CPU Intel i7-930, 2.80 GHz.
[d] IOLab [25], CPU Intel Core i7-7700, 4x3.60 GHz; mixed computation times; unrecorded number of runs.
[e] CPU Intel Core i7 12700F – 12x2.1 GHz; RAM 32 GB; OR-Tools CP-SAT 9.7 solver; 3600 sec time limit.

Table 6
Results on the larger instances from Kummer [28].

Instances	LR[a]	BRKGA[b]	BRKGA-MP-IPR[c]		MN-SA[d]	CP-SAT[e]	
	LB	UB	UB	Sec	UB	LB	UB
75_15_10_1.6_R_C	78.7	426.0	376.0	74.6	349.7	**163.9**	*357.7*
75_15_19_0.8_R_C	141.0	377.0	368.7	77.9	364.7	**210.0**	*381.0*
75_15_21_1.6_R_C	80.7	411.3	384.7	81.1	362.7	**122.8**	*399.7*
75_15_30_1.0_R_C	128.0	328.7	303.7	81.0	294.0	**176.3**	*313.3*
75_15_32_1.2_R_RC	151.7	380.0	353.7	76.3	337.3	**246.9**	*356.7*
75_15_35_0.8_R_C	130.3	774.3	719.0	72.9	703.7	**247.1**	*711.0*
75_15_35_1.0_C_RC	195.7	1145.3	1051.3	73.3	1033.3	**380.7**	*1043.7*
75_15_40_0.8_R_RC	184.3	452.7	423.7	82.9	408.3	**214.2**	*422.3*
75_15_5_0.8_C_RC	183.3	503.3	439.7	78.1	426.7	**263.6**	*443.7*
75_15_63_0.4_R_C	142.3	340.3	317.3	78.7	305.3	**185.5**	*317.0*
75_15_63_1.0_R_RC	147.3	453.3	408.0	78.2	392.7	**204.0**	*411.7*
75_15_65_1.0_C_C	118.7	280.0	257.0	78.8	252.3	**158.5**	*265.3*
75_15_79_1.2_R_C	89.3	495.0	404.3	79.1	394.7	**165.9**	*426.0*
75_15_84_0.8_R_RC	162.0	431.3	392.3	77.0	382.7	**264.1**	*401.0*
75_15_87_1.6_C_C	78.7	400.3	386.3	76.8	359.3	**165.1**	*378.3*
75_15_88_1.6_R_C	98.7	522.7	461.3	79.3	449.7	**166.6**	*456.7*
75_15_97_0.4_C_C	162.0	541.3	480.0	74.8	462.0	**293.8**	*465.7*
75_15_97_1.0_C_C	116.7	662.3	612.3	74.9	592.3	**211.9**	*597.0*
75_15_97_1.0_R_RC	144.4	924.3	798.0	72.5	780.0	**287.0**	*797.7*
75_15_98_1.6_R_C	81.5	423.3	375.7	76.7	358.7	**170.9**	*368.3*
Avg	130.8	513.6	465.7	77.2	450.5	**214.9**	*465.7*
100_20_10_1.2_R_C	132.7	340.3	300.0	103.6	298.0	**184.9**	*299.7*
100_20_11_1.2_R_C	118.0	634.0	535.3	95.2	508.3	**204.8**	*531.3*
100_20_23_1.2_R_C	102.0	528.7	454.0	107.6	437.0	**177.5**	*437.7*
100_20_25_0.8_R_C	138.3	750.7	654.7	97.8	624.0	**219.9**	*635.3*
100_20_30_1.0_R_C	133.3	707.0	621.7	101.7	602.3	**222.3**	*644.7*
100_20_35_1.6_C_RC	185.3	481.3	442.7	103.8	420.3	**252.0**	*433.3*
100_20_39_1.6_R_C	112.3	648.0	554.3	99.3	539.0	**191.3**	*561.0*
100_20_49_0.4_C_C	211.3	586.3	525.7	103.0	509.0	**312.9**	*519.7*
100_20_54_1.2_R_RC	176.0	435.3	398.7	101.4	400.3	**264.4**	*403.3*
100_20_57_1.6_R_C	105.0	574.7	505.7	105.0	485.3	**180.5**	*494.0*
100_20_61_0.4_R_RC	187.0	510.3	471.3	110.3	447.7	**286.4**	*456.7*
100_20_63_1.2_R_C	112.0	655.7	548.3	107.1	550.7	**202.2**	*559.3*
100_20_65_0.4_R_RC	212.0	533.7	495.0	104.8	481.3	**284.8**	*498.7*
100_20_68_1.6_R_C	87.0	351.3	304.7	115.4	304.7	**141.8**	*306.7*
100_20_76_1.6_R_C	100.7	568.0	500.0	98.4	470.0	**177.9**	*495.3*
100_20_77_1.0_R_C	134.7	730.0	613.7	105.6	604.0	**224.5**	*610.7*
100_20_88_1.2_C_C	111.3	301.7	269.7	99.4	264.0	**175.2**	*272.0*
100_20_8_1.2_R_C	117.0	729.7	607.3	105.7	604.7	**186.5**	*640.7*
100_20_99_0.4_C_RC	216.3	543.7	487.0	107.9	468.3	**300.3**	**465.0**
100_20_9_1.0_R_R	237.7	625.3	554.7	110.9	547.7	**346.2**	*558.3*
Avg	146.5	561.8	492.2	104.2	478.3	**226.8**	*491.2*
200_40_100_0.4_C_C	268.3	824.3	711.7	342.1	683.0	268.3	-
200_40_10_1.6_C_RC	309.0	1002.3	843.3	323.5	791.3	309.0	-
200_40_17_1.6_C_C	158.0	650.0	569.7	343.1	541.7	158.0	-
200_40_17_1.6_R_C	170.7	1005.0	835.0	340.2	783.7	**180.3**	-
200_40_30_1.6_R_C	180.3	1143.0	923.7	280.8	870.3	**199.2**	-
200_40_40_1.0_R_C	200.7	1128.3	913.0	297.2	849.0	**203.7**	-
200_40_51_1.0_R_C	214.0	1317.3	1136.7	292.7	1053.3	214.0	-
200_40_52_1.6_R_C	171.7	1109.7	932.0	277.0	871.0	**189.3**	-
200_40_60_0.4_C_RC	306.3	1005.0	870.7	268.9	827.3	**312.0**	-
200_40_69_1.0_C_R	342.7	1136.0	991.7	341.9	939.7	**345.0**	-
200_40_71_1.6_C_C	182.3	511.7	458.7	298.0	436.7	**184.0**	-
200_40_71_1.6_R_C	182.0	1018.3	857.7	274.8	790.7	**203.0**	-
200_40_75_1.6_R_C	200.3	791.7	679.0	296.4	643.0	**206.2**	-
200_40_76_1.2_R_C	218.0	1352.0	1133.3	301.4	1071.3	**238.5**	-
200_40_7_1.2_R_C	187.0	1207.7	1081.3	263.2	961.0	**195.0**	-
200_40_7_1.6_R_C	149.7	923.0	847.7	263.3	791.0	**165.6**	-
200_40_80_1.0_C_C	214.7	756.7	635.3	372.7	606.7	**218.0**	-
200_40_82_1.0_C_R	329.0	1123.0	944.0	296.2	894.0	**334.3**	-
200_40_98_1.2_R_C	202.0	1114.3	936.0	314.5	908.0	**209.0**	-
200_40_98_1.6_C_RC	294.0	944.3	830.3	311.0	782.7	294.0	-
Avg	224.0	1003.2	856.5	304.9	804.8	**231.3**	-

[a] Kummer [28], CPU Intel i7-930, 2.80 GHz, IBM CPLEX 20.1.0.0 solver; unreported time.
[b] Kummer [28], CPU Intel i7-930, 2.80 GHz; unreported time.
[c] Kummer [28], CPU Intel i7-930, 2.80 GHz.
[d] IOLab [25], CPU Intel Core i7-7700, 4x3.60 GHz; mixed computation times; unrecorded number of runs.
[e] CPU Intel Core i7 12700F – 12x2.1 GHz; RAM 32 GB; OR-Tools CP-SAT 9.7 solver; 3600 sec time limit.

Table 7
Results on the instances from IOLab [25].

Instances	MN-SA[a]	CP-SAT[b]	
	UB	LB	UB
instance_000-cosenza-r26-p299-s4-sim1.6-seq5.7	2016.3	**737.8**	-
instance_001-florence-r19-p165-s3-sim20.3-seq22.1	849.7	**385.3**	-
instance_002-venice-r11-p229-s4-sim23.9-seq24.1	863.3	**370.4**	-
instance_003-rome-r19-p44-s4-sim22.3-seq22.9	365.7	**301.1**	*376.0*
instance_004-venice-padua-treviso-r26-p258-s3-sim14.0-seq4.2	1253.3	**554.0**	-
instance_005-macerata-r29-p212-s2-sim9.1-seq7.4	1217.7	**478.3**	-
instance_006-cosenza-r30-p213-s5-sim20.5-seq8.4	1889.7	**669.3**	-
instance_007-venice-r29-p297-s3-sim2.9-seq7.1	1549.3	**571.3**	-
instance_008-cesena-r32-p247-s4-sim12.3-seq20.5	1569.7	**609.6**	-
instance_009-reggio-emilia-r15-p55-s2-sim21.7-seq7.6	297.7	**233.8**	*308.0*
instance_010-milan-r15-p76-s2-sim17.7-seq13.7	422.0	**274.8**	*448.7*
instance_011-perugia-r33-p181-s4-sim5.3-seq11.4	1190.3	**452.9**	-
instance_012-cesena-r37-p130-s2-sim4.0-seq20.9	911.7	**462.9**	-
instance_013-macerata-r15-p131-s5-sim22.9-seq13.2	659.7	**292.8**	-
instance_014-perugia-r29-p213-s3-sim20.1-seq22.5	1267.3	**500.7**	-
instance_015-cesena-r15-p73-s2-sim20.2-seq15.8	459.0	**242.5**	*524.7*
instance_016-macerata-r11-p145-s3-sim14.2-seq0.5	495.3	**243.6**	-
instance_017-rome-r26-p101-s3-sim9.8-seq3.7	526.0	**364.3**	*532.0*
instance_018-udine-r17-p356-s3-sim21.2-seq21.7	1412.3	**499.9**	-
instance_019-cesena-r18-p203-s4-sim21.7-seq21.3	1217.7	**471.2**	-
instance_020-cesena-r15-p78-s3-sim23.4-seq24.3	431.0	**267.8**	*454.0*
instance_021-florence-r26-p323-s3-sim18.1-seq12.6	1796.0	**696.3**	-
instance_022-reggio-emilia-r29-p255-s2-sim10.7-seq15.5	1565.0	**584.0**	-
instance_023-udine-r15-p75-s3-sim8.1-seq23.0	333.3	**265.6**	*338.3*
instance_024-perugia-r25-p270-s2-sim4.2-seq8.1	1267.7	**570.0**	-
instance_025-cesena-r18-p45-s5-sim18.9-seq12.6	487.7	**247.3**	*501.0*
instance_026-cosenza-r17-p191-s2-sim1.3-seq18.0	1006.7	**427.4**	-
instance_027-venice-padua-treviso-r10-p157-s4-sim24.3-seq23.8	595.3	**251.3**	-
instance_028-venice-padua-treviso-r32-p378-s4-sim4.6-seq14.7	2122.7	**703.0**	-
instance_029-macerata-r21-p100-s3-sim1.5-seq2.2	611.3	**372.0**	*650.3*
Avg	1021.7	**436.7**	-

[a] IOLab [25], CPU Intel Core i7-7700, 4x3.60 GHz; mixed computation times; unrecorded number of runs.
[b] CPU Intel Core i7 12700F – 12x2.1 GHz; RAM 32 GB; OR-Tools CP-SAT 9.7 solver; 3600 sec time limit.

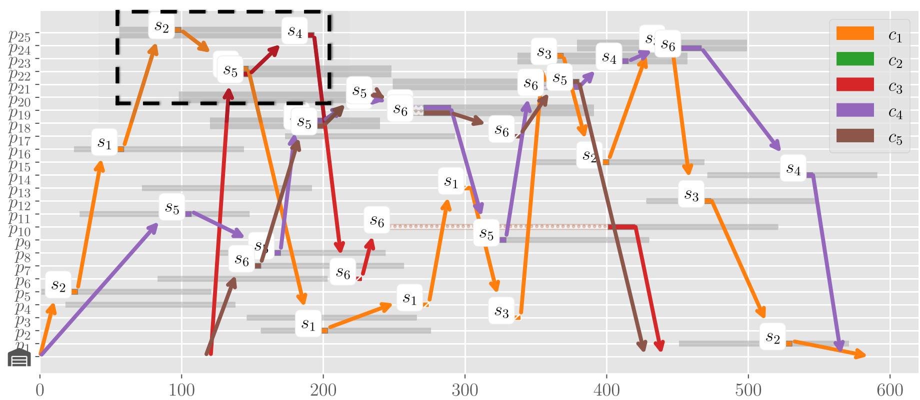

Fig. 2. Solution of instance B_6.

In this paper, a new model, significantly simplified with respect to those previously appeared, and based on the splitting of services required by a same patient into virtual copies of the patient itself, is presented and described in terms of constraint programming. An extensive experimental campaign to show the merits of the new model, carried out on datasets widely recognized by the research community, is presented. The comparison with the state-of-the-art methods in the literature suggests that the new model we propose is able to improve the best-known lower bounds consistently, and sometimes considerably. Solving the new model also leads to a few new best-known heuristic solutions, and in general to high-quality solutions for most of the instances (not on the largest ones, that are currently out-of-reach). This is remarkable given the simplicity behind the model, and the limited implementation effort required

Table 8
Comparison between the state-of-the-art MILP solver Gurobi and CP-SAT on some representative instances.

Instances	Gurobi (MILP)[a]		CP-SAT[b]	
	LB	UB	LB	UB
from Mankowska et al. [35]				
B6 (25 patients)	**440.9**	440.9	*393.1*	440.9
B9 (25 patients)	**350.0**	402.7	*319.3*	402.7
C1 (50 patients)	*491.4*	*948.2*	**499.0**	**943.7**
C3 (50 patients)	**433.2**	*547.0*	*419.8*	**537.8**
D2 (75 patients)	*416.2*	*725.5*	**461.5**	**653.3**
D4 (75 patients)	*452.3*	*822.5*	**453.5**	**772.4**
E1 (100 patients)	*489.1*	*5330.8*	**572.3**	**1275.4**
E6 (100 patients)	*429.3*	*856.1*	**435.7**	**766.8**
F2 (200 patients)	*547.9*	-	**582.2**	-
F4 (200 patients)	*538.2*	-	**547.7**	-
G6 (300 patients)	*548.3*	-	**613.1**	-
G9 (300 patients)	*588.2*	-	**699.5**	-
from Kummer [28]				
25_5_22_1.0_C_C	**229.7**	820.3	*217.8*	820.3
50_10_53_1.2_R_C	**178.7**	*265.0*	*166.6*	**251.0**
50_10_80_1.0_C_C	**245.4**	*870.0*	*241.0*	**815.7**
75_15_40_0.8_R_RC	*209.1*	-	**214.2**	**422.3**
100_20_30_1.0_R_C	*212.1*	-	**222.3**	**644.7**
100_20_68_1.6_R_C	*139.5*	-	**141.8**	**306.7**
200_40_51_1.0_R_C	*207.8*	-	**214.0**	-

[a] CPU Intel Core i7 12700F – 12x2.1 GHz; RAM 32 GB; Gurobi 11.0 solver; 3600 sec time limit.
[b] CPU Intel Core i7 12700F – 12x2.1 GHz; RAM 32 GB; OR-Tools CP-SAT 9.7 solver; 3600 sec time limit.

to have it in operation. Finally, considerations about the intrinsic limitations behind the best existing metaheuristic approaches, and their impacts on the results have been reported.

Future research could involve the embedding of the new constraint programming model within metaheuristic methods, leading to hybrid methods capable of retaining the best of the approaches. Namely, the capability of the model of producing lower bounds and carrying out domain-independent heuristic searches, and the capability of metaheuristics to search the solution space with a HHC-oriented vision. Finally, we plan to extend our model in order to apply it to other variants of the HHC problem. For example, the present formulation does not consider any penalty for the waiting time of caregivers at patients location due to early arrivals, and this aspect might be of extreme interest for real-world applications. Furthermore, it does not consider the possibility that a caregiver goes to the first patient directly from home, without passing through the central office first. This would lead to a multi-depot problem, and again would fit well with realistic settings.

CRediT authorship contribution statement

Roberto Montemanni: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis. **Sara Ceschia:** Writing – review & editing, Writing – original draft, Visualization, Validation, Resources, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Andrea Schaerf:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We thank Luca Di Gaspero for developing the viewer used for creating Figs. 1 and 2.

Sara Ceschia has been funded by the Italian “Ministero dell’Università e della Ricerca” – *NextGenerationEU*, under the project “Modeling and solving a real-world home healthcare routing and scheduling problem”.

Appendix A

In this appendix we aim at comparing a state-of-the-art MILP solver operating on the model presented in Section 4 and the CP-SAT solver working on the same model, but with the improvements discussed in Section 4.1. A set of 19 representative instances has been selected at random for the comparison. Please refer to Section 5 for a description of the instances. The state-of-the-art solvers Gurobi 11.0 [22] and CP-SAT 9.7 [44] are used, and all the tests reported are executed on the same machine, with the same maximum execution time of 3600 seconds. The results are summarized in Table 8 in terms of lower and upper bounds retrieved in the given time. Entries in bold highlight that the respective solver was better, entries in italics indicate that the respective solver was worse.

The results of Table 8 indicate that Gurobi is more effective in producing lower bounds for the smaller problems, although the differences between the two solver remain marginal. When the size of the instances increases, CP-SAT produces better lower bounds, and sometimes the gap is substantial. In terms of upper bounds, CP-SAT is consistently better than Gurobi, with a draw only for the three smallest instances.

We can conclude that for the particular compact model we propose, CP-SAT seems to currently perform better than Gurobi on the significant instances (the smallest instances are considered not very relevant, since most of them can be already solved to optimality), and therefore we used such a solver as a reference one.

Notice that the instance B6 from Mankowska et al. [35] was closed for the first time by Gurobi in these experiments.

References

[1] S.R. Ait Haddadene, N. Labadie, C. Prodhon, A GRASP × ILS for the vehicle routing problem with time windows, synchronization and precedence constraints, Expert Syst. Appl. 66 (2016) 274–294.

[2] M. Bazirha, R. Benmansour, A. Kadrani, An efficient two-phase heuristic for the home care routing and scheduling problem, Comput. Ind. Eng. 181 (2023) 109329.

[3] M. Bazirha, A. Kadrani, R. Benmansour, Stochastic home health care routing and scheduling problem with multiple synchronized services, Ann. Oper. Res. 320 (2) (2023) 573–601.

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

[5] S. Bertels, T. Fahle, A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem, Comput. Oper. Res. 33 (10) (2006) 2866–2890.

[6] D. Bredström, M. Rönnqvist, Combined vehicle routing and scheduling with temporal precedence and synchronization constraints, Eur. J. Oper. Res. 191 (1) (2008) 19–31.

[7] S. Ceschia, L. Di Gaspero, R.M. Rosati, A. Schaerf, Multi-neighborhood simulated annealing for the home healthcare routing and scheduling problem, https://www.researchsquare.com/article/rs-4086164/v1, 2024.

[8] S. Ceschia, L. Di Gaspero, A. Schaerf, Simulated annealing for the home healthcare routing and scheduling problem, in: Advances in Artificial Intelligence, Springer, 2022, pp. 402–412.

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

[10] M. Cissé, S. Yalçındağ, Y. Kergosien, E. Şahin, C. Lenté, A. Matta, OR problems related to home health care: a review of relevant routing and scheduling problems, Oper. Res. Health Care 13 (2017) 1–22.

[11] Y. Clapper, J. Berkhout, R. Bekker, D. Moeke, A model-based evolutionary algorithm for home health care scheduling, Comput. Oper. Res. 150 (2023) 106081.

[12] J. Decerle, O. Grunder, A.H. El Hassani, O. Barakat, A memetic algorithm for a home health care routing and scheduling problem, Oper. Res. Health Care 16 (2018) 59–71.

[13] M. Dell'Amico, J. Jamal, R. Montemanni, Modelling and solving the precedence-constrained minimum-cost arborescence problem with waiting-times, in: Proceedings of the 18th Conference on Computer Science and Intelligence Systems, IEEE, 2023, pp. 421–430.

[14] M. Dell'Amico, J. Jamal, R. Montemanni, Compact models to solve the precedence-constrained minimum-cost arborescence problem with waiting times, Algorithms 17 (1) (2024) 12.

[15] L. Di Gaspero, T. Urli, A CP/LNS approach for multi-day homecare scheduling problems, in: Hybrid Metaheuristics, Springer, 2014, pp. 1–15.

[16] M. Di Mascolo, C. Martinez, M.-L. Espinouse, Routing and scheduling in home health care: a literature survey and bibliometric analysis, Comput. Ind. Eng. 158 (2021) 107255.

[17] P. Eveborn, P. Flisberg, M. Ronnqvist, Laps care—an operational system for staff planning of home care, Eur. J. Oper. Res. 171 (3) (2006) 962–976.

[18] C. Fikar, P. Hirsch, Home health care routing and scheduling: a review, Comput. Oper. Res. 77 (2017) 86–95.

[19] World Health Organization, in: N. Genet, W. Boerma, M. Kroneman, A. Hutchinson, R.B. Saltman (Eds.), Homecare Across Europe, European Observatory on Health Systems and Policies, 2012.

[20] Google, OR-Tools, https://developers.google.com/optimization/, 2023. (Accessed 14 November 2023).

[21] F. Grenouilleau, A. Legrain, N. Lahrichi, L.M. Rousseau, A set partitioning heuristic for the home health care routing and scheduling problem, Eur. J. Oper. Res. 275 (1) (2019) 295–303.

[22] Gurobi Optimization, LLC, Gurobi Optimizer Reference Manual, 2023. (Accessed 1 July 2023).

[23] G. Hiermann, M. Prandtstetter, A. Rendl, J. Puchinger, G.R. Raidl, Metaheuristics for solving a multimodal home-healthcare scheduling problem, Cent. Eur. J. Oper. Res. 23 (2015) 89–113.

[24] IBM ILOG, User's manual for CPLEX, https://www.cplex.com/, 2023. (Accessed 1 July 2023).

[25] Intelligent Optimization Laboratory (IOLab), Università degli Studi di Udine, Data and toolbox repository for the home healthcare routing and scheduling problem, https://github.com/iolab-uniud/hhcrsp, 2023. (Accessed 14 November 2023).

[26] G. Kordi, A. Divsalar, S. Emami, Multi-objective home health care routing: a variable neighborhood search method, Optim. Lett. (2023) 1–42.

[27] A.F. Kummer, A new curated benchmark dataset for the home health care problem, https://github.com/afkummer/hhcrsp-dataset-2021, 2021. (Accessed 14 November 2023).

[28] A.F. Kummer, A study on the home care routing and scheduling problem, PhD thesis, Universidade Federal do Rio Grande do Sul, 2021.

[29] A.F. Kummer, L.S. Buriol, O.C. de Araújo, 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, 2020, pp. 717–724.

[30] A.F. Kummer, O.C.B. de Araújo, L.S. Buriol, M.G.C. Resende, A biased random-key genetic algorithm for the home health care problem, Int. Trans. Oper. Res. 31 (3) (2024) 1859–1889.

[31] S. Lasfargeas, C. Gagné, A. Sioud, Solving the home health care problem with temporal precedence and synchronization, in: Bioinspired Heuristics for Optimization, Springer, 2019, pp. 251–267.

[32] R. Liu, B. Yuan, Z. Jiang, Mathematical model and exact algorithm for the home care worker scheduling and routing problem with lunch break requirements, Int. J. Prod. Res. 55 (2) (2017) 558–575.

[33] R. Liu, B. Yuan, Z. Jiang, A branch-and-price algorithm for the home-caregiver scheduling and routing problem with stochastic travel and service times, Flex. Serv. Manuf. J. 31 (2019) 989–1011.

[34] W. Liu, Ma. Dridi, H. Fei, A.H. El Hassani, Hybrid metaheuristics for solving a home health care routing and scheduling problem with time windows, synchronized visits and lunch breaks, Expert Syst. Appl. 183 (2021) 115307.

[35] D.S. Mankowska, F. Meisel, C. Bierwirth, The home health care routing and scheduling problem with interdependent services, Health Care Manage. Sci. 17 (1) (2014) 15–30.

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

[37] R. Montemanni, M. Dell'Amico, Constraint programming models for the parallel drone scheduling vehicle routing problem, EURO J. Comput. Optim. 11 (2023) 100078.

[38] R. Montemanni, M. Dell'Amico, Solving the parallel drone scheduling traveling salesman problem via constraint programming, Algorithms 16 (1) (2023) 40.

[39] R. Montemanni, M. Dell'Amico, A. Corsini, Parallel drone scheduling vehicle routing problems with collective drones, Comput. Oper. Res. 163 (2024) 106514.

[40] T.V.L. Nguyen, R. Montemanni, Mathematical programming models for home health care service optimization, Int. J. Oper. Res. 25 (4) (2016) 449–463.

[41] University of Kiel, Research data, https://www.scm.bwl.uni-kiel.de/de/forschung/research-data, 2021. (Accessed 14 November 2023).

[42] N. Oladzad-Abbasabady, R. Tavakkoli-Moghaddam, M. Mohammadi, B. Vahedi-Nouri, A bi-objective home care routing and scheduling problem considering patient preference and soft temporal dependency constraints, Eng. Appl. Artif. Intell. 119 (2023) 105829.

[43] S.N. Parragh, K.F. Doerner, Solving routing problems with pairwise synchronization constraints, Cent. Eur. J. Oper. Res. 26 (2018) 443–464.

[44] L. Perron, V. Furnon, CP-SAT, https://developers.google.com/optimization/cp/cp_solver, 2024. (Accessed 14 March 2024).

[45] M.S. Rasmussen, T. Justesen, A. Dohn, J. Larsen, The home care crew scheduling problem: preference-based visit clustering and temporal dependencies, Eur. J. Oper. Res. 219 (3) (2012) 598–610.

[46] A. Rendl, M. Prandtstetter, G. Hiermann, J. Puchinger, G. Raidl, Hybrid heuristics for multimodal homecare scheduling, in: Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, Springer, 2012, pp. 339–355.

[47] P.J. Stuckey, Lazy clause generation: combining the power of SAT and CP (and MIP?) solving, in: Proceedings of the International Conference on Integration of Artificial Intelligence and Operations Research Techniques in Constraint Programming, CPAIOR, 2010, pp. 5–9.

[48] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67.

[49] T. Xiang, Y. Li, W.Y. Szeto, The daily routing and scheduling problem of home health care: based on costs and participants' preference satisfaction, Int. Trans. Oper. Res. 30 (1) (2023) 39–69.