Health Care Manag Sci (2014) 17:15–30
DOI 10.1007/s10729-013-9243-1

The home health care routing and scheduling problem with interdependent services

Dorota Slawa Mankowska · Frank Meisel · Christian Bierwirth

Received: 23 November 2012 / Accepted: 20 May 2013 / Published online: 19 June 2013

Abstract This paper presents a model for the daily planning of health care services carried out at patients' homes by staff members of a home care company. The planning takes into account individual service requirements of the patients, individual qualifications of the staff and possible interdependencies between different service operations. Interdependencies of services can include, for example, a temporal separation of two services as is required if drugs have to be administered a certain time before providing a meal. Other services like handling a disabled patient may require two staff members working together at a patient's home. The time preferences of patients are included in terms of given time windows. In this paper, we propose a planning approach for the described problem, which can be used for optimizing economical and service oriented measures of performance. A mathematical model formulation is proposed together with a powerful heuristic based on a sophisticated solution representation.

Keywords Home health care · Routing and scheduling · Double services · Temporal interdependencies

1 Introduction

The World Health Organization has announced that the rate of care-dependent elderly people in Europe will strongly increase in the next decades, see [13]. In Germany, the number of care-dependent people has already reached 1.6 million in 2012 and is expected to grow further, see [4]. This situation is caused by several factors like, for example, the demographic change due to low birthrates and increasing life expectation. In addition, family members more often live apart from each other, for example, because children move to other cities in order to find jobs while the parents stay in the hometown. Such situation complicates providing daily care to elderly relatives. Nowadays, providing care is shifted more and more to Home Care Companies (HCCs). These companies offer a broad range of on-site services to persons in need in order to relieve family, hospitals, and nursing homes. For this purpose, they employ differently qualified staff, including nurses, therapists, social workers, physicians, food couriers, and the like. The individual range of qualifications comprises, for example, a number of languages spoken, permission to run errands, or a license to administer drugs. The staff members (also further referred to as caregivers or employees) are typically equipped with cars or bikes, enabling them to move among patients' homes and the HCC's central office. The patients require certain types of services, which must be performed by suitably qualified staff members, preferably within a given time window.

We distinguish in this paper *single services* and *double services*. A single service consists of one service operation to be executed by a single staff member. A double service consists of two service operations that are performed by two staff members. Double services are further divided into simultaneous services and services with a given precedence relation. A simultaneous service is necessary if, for example, lifting a disabled person requires two staff members. Services with a precedence relation occur if, for example, drugs must be administered a certain time before providing a meal. Such a requirement defines a time distance between

D. S. Mankowska (✉) · F. Meisel · C. Bierwirth
Martin-Luther-University Halle-Wittenberg, Grosse Steinstrasse 73, 06108 Halle, Germany
e-mail: dorota.mankowska@wiwi.uni-halle.de

F. Meisel
e-mail: frank.meisel@wiwi.uni-halle.de

C. Bierwirth
e-mail: christian.bierwirth@wiwi.uni-halle.de

two operations, which must be kept to achieve a medical service of desired quality as is mandatory in the HCC business. Since there is a lack of planning tools that can handle such services, HCC planners schedule double services by hand, which leads to poor-quality work plans, cf. [5, 12]. In [7, 12] practical case data is reported with up to 10 % of the patients requiring double services. At the same time, Rasmussen et al. [12] conjecture the actual number of double services being even higher because the manual planning of double services makes the real world data partially incomplete. The above verifies that such services are relevant in the daily operations of HCCs and that there is a lack of decision support.

The planning problem considered in this paper is to route staff members and to schedule single and double service operations for the HCC according to the individual service requirements of a given set of patients. We call this problem the Home Health Care Routing and Scheduling Problem (HHCRSP). The problem differs from the well-known multiple Traveling Salesman Problem with Time Windows (mTSPTW) in the following aspects: First, in the HHCRSP there exist customers that must be visited more than once. Second, the caregivers possess different skills and qualifications. Third, temporal interdependencies of double services must be taken into consideration. The latter necessitates a careful synchronization of the interdependent working plans of the staff.

The contribution of this paper is to provide a mathematical formulation for the HHCRSP, a new solution representation that can accommodate single services and double services, and powerful solution methods like, for example, an Adaptive Variable Neighborhood Search (AVNS). In particular, we consider three objectives. The first is to minimize the total daily distance traveled by the staff, which corresponds to a minimization of the total traveling cost. The second is to minimize the tardiness of services to avoid unnecessary waiting times of the patients. The third goal is to provide a fair allocation of unavoidable waiting times among the patients. The fairness goal is achieved by minimizing the largest observed tardiness among all services.

The outline of this paper is as follows. In Section 2, we survey related studies. A mathematical formulation of the HHCRSP is presented in Section 3 together with an illustrative example. Section 4 presents the heuristic solution methods, which take advantage from a new solution representation. Numerical experiments are conducted in Section 5. The paper is concluded in Section 6.

2 Literature

Investigating the HHCRSP is a young but growing research field. Different variants of problem formulations have been studied in the literature so far. The main differences among the considered problems are (1.) whether to assume a homogeneous team of staff members or a heterogeneous one, (2.) whether to consider synchronization of interdependent services or not, (3.) the particular objectives pursued in the planning, and (4.) the methodology used for solving the resulting optimization problems.

Table 1 presents papers dealing with the HHCRSP together with their main characteristics. Homogeneous staff members are considered in three papers. Akjiratikarl et al. [1] present an application of Particle Swarm Optimization to home care employee scheduling in the UK without considering interdependent services. Bredström and Rönnqvist [7] model home care staff scheduling as a routing problem for homogeneous vehicles with precedence and synchronization constraints. Rasmussen et al. [12] propose a set-partitioning approach and additional side constraints, which are used within a branch-and-price algorithm for solving the problem.

The further papers take heterogeneous staff members into consideration. Begur et al. [2] were the first who developed a decision support system for home health care companies. They aim at balancing the workload of employees and propose construction and improvement heuristics to solve the problem. Cheng and Rich [8] present a mixed-integer linear programming model (MILP) for the problem and propose simple solution heuristics. Here, the goal is to minimize the cost associated with overtime and part-time work. Bertels and Fahle [3] propose to combine linear programming, constraint programming and metaheuristics for solving an HHCRSP. Bräysy et al. [6] consider a case study for home care services from Finland. In different scenarios, the routing of HCC employees is determined by a commercial solver. Trautsamwieser and Hirsch [15] propose a model formulation and a Variable Neighborhood Search for optimizing the daily scheduling of nurses. In all of these papers, interdependent services are not considered explicitly.

Interdependent services are subject of two other papers. Eveborn et al. [9] present an operational system for staff planning of HCCs called LAPS CARE. A set-partitioning model is proposed, where the patients are divided into subgroups and assigned to the routes of the available staff. A repeated matching heuristic is presented for solving instances with up to 123 patients and 21 employees. The order of visiting the patients is determined using a nearest neighbor heuristic that also respects time windows. Interdependent visits by one patient are split into two visits and the start times of these visits are fixed. The idea of the solution approach is to penalize routes, where some patients are not served at all. Kergosien et al. [11] view a home care problem as an extended multiple Traveling Salesman Problem. Multiple visits at one patient are represented by duplicating nodes. The task is to find routes with all nodes visited

Table 1 Literature on home health care routing and scheduling

Paper	Year	Staff	Interdependent services	Performance measure (objective)	Solution method
[1]	2007	hom	No	Distance traveled	Particle Swarm Optimization
[7]	2008	hom	Yes	Weighted sum of preferences, traveling time	MILP Solver, local branching heuristic
[12]	2012	hom	Yes	Weighted sum of uncovered visits, visit disfavors, and total traveling costs	Branch-and-Price
[2]	1997	het	No	Workload balancing and traveling time	k-opt, nearest neighbor
[8]	1998	het	No	Cost associated with overtime and part-time work	MILP Solver, route construction and improvement heuristics
[3]	2006	het	No	Distance traveled, penalties for violated time windows and qualification requirements	Combination of linear programming, constraint programming, Tabu Search and Simulated Annealing
[6]	2007	het	No	Distance traveled, working time of nurses	Commercial VRP solver
[15]	2011	het	No	Distance traveled, dissatisfaction of clients and nurses	Variable Neighborhood Search
[9]	2006	het	Yes	Distance traveled, traveling costs, scheduled hours, inconvenient working hours, preferences	Set-partitioning model, repeated matching algorithm
[11]	2009	het	Yes	Distance traveled	MILP Solver
This paper		het	Yes	Distance traveled, violation of time windows, service fairness among all patients	Construction heuristic, local search, Adaptive Variable Neighborhood Search

exactly once. Visits at the nodes representing interdependent services have to respect a temporal constraint. The authors divide the staff into subgroups that represent certain qualification levels like nurses, therapists etc. A MILP is solved by a standard solver. To enhance the solver, additional cuts and other technical improvements are developed. This way, instances with up to 40 services and a team of three staff members can be solved exactly. No solution method is proposed for larger instances that cannot be tackled by the solver.

From Table 1 we observe that most studies consider more than one performance measure in the optimization, typically by means of a linear combination. This is because HCCs are service providers who have to plan their operations with respect to both the cost of providing services to patients and the quality of the performed services. The implementation of the cost measures is often done in terms of minimizing the distances traveled by caregivers [1, 3, 6, 9, 11, 15] or by minimizing the traveling times [2, 7]. Service quality is considered in diverse dimensions, including for example the timely fulfillment of services [3, 9, 15], compliance with preferences regarding the assignment of certain caregivers to certain patients [7, 9, 12], the minimization of unfulfilled services [12], or the minimization of violated qualification requirements [3].

The contribution of this paper is a new mathematical formulation for the HHCRSP which overcomes limitations observed in recent models. This includes combining the features of heterogeneous staff, temporal interdependencies among services, and balancing cost and service objectives. Also, no unfulfilled services are allowed and all patients must be visited by respectively qualified caregivers. To achieve all patients receiving their services, we allow operations being tardy. The model supports both types of interdependent services, i. e. patients can require simultaneous services or services with given precedence relation. As solving this rich model is very difficult, a second contribution of this paper is providing new solution methods that are capable of solving very large instances with several hundred patients within minutes.

3 Modeling the HHCRSP

In the following, we propose a mathematical formulation for the HHCRSP. We first present the used notation followed by the optimization model and a numerical example.

3.1 Notation

A set of patients $\mathcal{C}$, a set of service types $\mathcal{S}$, and a staff $\mathcal{V}$ of the HCC are given. The qualifications of the staff members are expressed by a binary parameter a_{vs}, which is equal to 1 if staff member $v \in \mathcal{V}$ is qualified to perform a service of type $s \in \mathcal{S}$. The employees start their routes at the HCC's central office, which refers to node 0. The set of patients' locations and the central office form the node set $\mathcal{C}^0 = \mathcal{C} \cup \{0\}$. The set of patients $\mathcal{C}$ is divided into patients $\mathcal{C}^{\mathrm{s}}$ requiring a single service and patients $\mathcal{C}^{\mathrm{d}}$ requiring a double service. The service requirements of each patient $i \in \mathcal{C}$ are denoted by parameter r_{is}, with $r_{is} = 1$ if patient i requires a service of type s, and 0 otherwise. Furthermore, for double service patients $i \in \mathcal{C}^{\mathrm{d}}$, a minimal time distance $\delta_i^{\min}$ and a maximal time distance $\delta_i^{\max}$ are specified between the first and the second service. If $\delta_i^{\min} = \delta_i^{\max} = 0$, both service operations have to start simultaneously. Otherwise, for $\delta_i^{\min} > 0$, the second service operation must start at least $\delta_i^{\min}$ time units after the start of the first service operation. Moreover, the second service must not start later than $\delta_i^{\max}$ time units after the first service $\left(\delta_i^{\max} \geq \delta_i^{\min}\right)$. It is required that the minimal and maximal time distance between the two operations in a double service are always respected. We assume that services are indexed such that the first service required by a patient always has a lower index s than the second service. Based on $\delta_i^{\min}$ and $\delta_i^{\max}$, we further divide double service patients into patients $\mathcal{C}^{\mathrm{sim}}$ with simultaneous service $\left(\delta_i^{\min} = \delta_i^{\max} = 0\right)$ and into patients $\mathcal{C}^{\mathrm{prec}}$ with a precedence among the service operations $\left(0 < \delta_i^{\min} \leq \delta_i^{\max}\right)$. Note that patients requiring multiple services which are independent from each other are modeled as two single service patients. To the contrary, double service patients require two qualified services that strictly depend on each other. To avoid cycles in the routes of caregivers, we assume that the two operations of a double service are to be executed by different caregivers.

The distance between any two locations i and j is denoted by d_{ij}. We assume that the traveling time from location i to location j is proportional to the distance d_{ij}. Service durations p_{is} denote the time needed to perform service $s \in \mathcal{S}$ at patient $i \in \mathcal{C}$. Moreover, for each patient $i \in \mathcal{C}$, a time window $[e_i, l_i]$ is given for the start of its service operations. In case that an employee arrives at patient i before time e_i, he or she has to wait until the time window opens. If the staff member arrives at a single service patient $i \in \mathcal{C}^{\mathrm{s}}$ after time e_i, the service is started immediately. Double services with simultaneous operations are also started immediately, provided that the other caregiver has already arrived at the patient's home. Service operations with a precedence relation start according to the given time distances $\delta_i^{\min}$ and $\delta_i^{\max}$.

The routing of the staff is modeled by binary decision variables x_{ijvs}, which take value 1, if caregiver $v \in \mathcal{V}$ moves directly from $i \in \mathcal{C}^0$ to $j \in \mathcal{C}^0$ to provide service operation $s \in \mathcal{S}$ for patient j, and 0 otherwise. The scheduling variable t_{ivs} denotes the start time of the service operation $s \in \mathcal{S}$ at patient $i \in \mathcal{C}$ if served by staff member $v \in \mathcal{V}$. In case that a service s starts beyond the end of the time window l_i, a tardiness value z_{is} is gathered by the time span between t_{ivs} and l_i. The entire notation is summarized in Table 2.

Table 2 Notation used for modeling the HHCRSP

Data	
$\mathcal{C}$	Set of all patients
$\mathcal{S}$	Set of offered service types
$\mathcal{V}$	Set of staff members
a_{vs}	Equal 1 iff employee $v \in \mathcal{V}$ is qualified to provide service operation $s \in \mathcal{S}$
r_{is}	Equal 1 iff patient $i \in \mathcal{C}$ requires service operation s
$\mathcal{C}^0$	Set of all locations; $\mathcal{C}^0 = \mathcal{C} \cup \{0\}$
$\mathcal{C}^{\mathrm{s}}$	Set of single service patients
$\mathcal{C}^{\mathrm{d}}$	Set of double service patients; $\mathcal{C}^{\mathrm{d}} = \mathcal{C}^{\mathrm{sim}} \cup \mathcal{C}^{\mathrm{prec}}$
$\mathcal{C}^{\mathrm{sim}}$	Set of patients requiring a simultaneous double service
$\mathcal{C}^{\mathrm{prec}}$	Set of patients requiring a double service with precedence
$\delta_i^{\min}$	Minimal time distance between service start times at patient $i \in \mathcal{C}^{\mathrm{d}}$
$\delta_i^{\max}$	Maximal time distance between service start times at patient $i \in \mathcal{C}^{\mathrm{d}}$
$[e_i, l_i]$	Time window of patient $i \in \mathcal{C}$
d_{ij}	Traveling distance between locations $i \in \mathcal{C}^0$ and $j \in \mathcal{C}^0$
p_{is}	Processing time of service operation s at patient $i \in \mathcal{C}$
Decision variables	
x_{ijvs}	Binary, 1 iff staff member v moves from i to j for providing service operation s, 0 otherwise
t_{ivs}	Start time of service operation s at patient i provided by staff member v
z_{is}	Tardiness of service operation s at patient i

3.2 MILP formulation

We consider three measures of performance, denoted by D, T, and $T^{\max}$:

- D measures the total distance traveled by all caregivers for visiting patients.
- T measures the total tardiness of services that start beyond the time windows.
- $T^{\max}$ measures the maximal tardiness observed over all service operations.

The idea behind the last measure is to avoid a situation, where a few patients accumulate a large amount of tardiness at the favor of the rest of the patients. The three measures are reflected in the combined and weighted objective function (1).

$$\min \rightarrow Z = \lambda_1 D + \lambda_2 T + \lambda_3 T^{\max} \tag{1}$$

This function can be parametrized flexibly such that an HCC can either pursue a minimization of cost, or a maximization of service quality, or a combination thereof. More precisely, if an HCC considers only one of the objectives relevant, the rest is ignored by setting the corresponding λ-values to 0. A hierarchy among the goals is expressible by setting strongly different values. For example, setting λ_1 to a very high value and λ_2 and λ_3 to much lower values gives priority to the minimization of the traveling times. In this situation, among all solutions with minimum travel effort, the one with lowest tardiness of services will be chosen. Finally, an economic tradeoff of cost and service quality is achieved if the λ-values are set to the cost rate per distance unit traveled by a caregiver and to penalty cost rates for each time unit of tardiness. In this case, the particular cost rates have to be specified by the HCC.

The performance measures D, T, and $T^{\max}$ are determined in constraints (2)–(4).

$$D = \sum_{v\in\mathcal{V}}\sum_{i\in\mathcal{C}^0}\sum_{j\in\mathcal{C}^0}\sum_{s\in\mathcal{S}} d_{ij}\cdot x_{ijvs} \tag{2}$$

$$T = \sum_{i\in\mathcal{C}}\sum_{s\in\mathcal{S}} z_{is} \tag{3}$$

$$T^{\max} \geq z_{is} \quad \forall i\in C, s\in S \tag{4}$$

Constraints (5) guarantee that the route of each staff member starts and ends in the HCC's central office.

$$\sum_{i\in\mathcal{C}^0}\sum_{s\in\mathcal{S}} x_{0ivs} = \sum_{i\in\mathcal{C}^0}\sum_{s\in\mathcal{S}} x_{i0vs} = 1 \qquad \forall v\in\mathcal{V} \tag{5}$$

Constraints (6) are inflow-outflow conditions, which ensure that an employee v, who visits patient i, has to leave this patient after the service.

$$\sum_{j\in\mathcal{C}^0}\sum_{s\in\mathcal{S}} x_{jivs} = \sum_{j\in\mathcal{C}^0}\sum_{s\in\mathcal{S}} x_{ijvs} \qquad \forall i\in\mathcal{C},\ v\in\mathcal{V} \tag{6}$$

Constraints (7) ensure that every required service operation s is assigned to exactly one qualified caregiver.

$$\sum_{v\in\mathcal{V}}\sum_{j\in\mathcal{C}^0} a_{vs}\cdot x_{jivs} = r_{is} \qquad \forall i\in\mathcal{C}, s\in\mathcal{S} \tag{7}$$

Constraints (8) determine the start times of the service operations with respect to service durations and traveling times. The constraints enforce that the start times of services along the route of a staff member are strictly increasing. In doing so, they also avoid cycles in the routes because a return to an already visited patient would violate the start time of the prior visit.

$$t_{ivs_1} + p_{is_1} + d_{ij} \leq t_{jvs_2} + M\left(1 - x_{ijvs_2}\right) \qquad \forall i\in\mathcal{C}^0, j\in\mathcal{C}, v\in\mathcal{V}, s_1, s_2\in\mathcal{S} \tag{8}$$

Note that the domain of (8) can be further restricted to those caregivers v that are qualified for services s_1 and s_2 (i. e., $a_{vs_1} = a_{vs_2} = 1$) and those services that are actually required by i and j (i. e., $r_{is_1} = r_{js_2} = 1$). This avoids establishing constraints that are not relevant in a particular problem instance.

Constraints (9) and (10) determine the compliance of service start times with the time windows. Note that the start time e_i imposes a hard constraint, whereas its end time may be overshot, which is expressed by a tardiness $z_{is} > 0$.

$$t_{ivs} \geq e_i \qquad \forall i\in\mathcal{C}, v\in\mathcal{V}, s\in\mathcal{S} \tag{9}$$

$$t_{ivs} \leq l_i + z_{is} \qquad \forall i\in\mathcal{C}, v\in\mathcal{V}, s\in\mathcal{S} \tag{10}$$

The temporal interdependencies of operations in double services are reflected in (11) and (12). The minimal time distance between the service start times is guaranteed by

$$t_{iv_2s_2} - t_{iv_1s_1} \geq \delta_i^{\min} - M\left(2 - \sum_{j\in\mathcal{C}^0} x_{jiv_1s_1} - \sum_{j\in\mathcal{C}^0} x_{jiv_2s_2}\right) \qquad \forall i\in\mathcal{C}^{\mathrm{d}}, v_1, v_2\in\mathcal{V}, s_1, s_2\in\mathcal{S}: s_1 < s_2 \tag{11}$$

The maximal time distance is guaranteed by

$$t_{iv_2s_2} - t_{iv_1s_1} \leq \delta_i^{\max} + M\left(2 - \sum_{j\in\mathcal{C}^0} x_{jiv_1s_1} - \sum_{j\in\mathcal{C}^0} x_{jiv_2s_2}\right) \qquad \forall i\in\mathcal{C}^{\mathrm{d}}, v_1, v_2\in\mathcal{V}, s_1, s_2\in\mathcal{S}: s_1 < s_2 \tag{12}$$

Similar to (8), the domains of (9) to (12) can be reduced to those combinations of services, caregivers and patients that are relevant in a problem. The domains of the routing variables x_{ijvs} are defined in (13). Here, $a_{vs}\cdot r_{js} = 1$, if and only if staff member $v\in\mathcal{V}$ is qualified to perform a service $s\in\mathcal{S}$ that is required by a patient $j\in\mathcal{C}$, and 0 otherwise. This restriction eliminates superfluous binary variables for patients whose service requirements are incompatible with the qualification of a caregiver.

$$x_{ijvs}\in\left\{0, a_{vs}\cdot r_{js}\right\} \qquad \forall i, j\in\mathcal{C}^0, v\in\mathcal{V}, s\in\mathcal{S} \tag{13}$$

Finally, non-negativity constraints define domains for the scheduling and tardiness variables.

$$t_{ivs},\ z_{is} \geq 0 \qquad \forall i\in\mathcal{C}^0, v\in\mathcal{V}, s\in\mathcal{S} \tag{14}$$

The described model contains the multiple Traveling Salesman Problem with Time Windows (mTSPTW) as a special case. This becomes obvious by considering a problem setting with only one type of service ($|\mathcal{S}| = 1$), no double service patients ($\mathcal{C}^{\mathrm{d}} = \emptyset; \mathcal{C} = \mathcal{C}^{\mathrm{s}}$) and a homogeneous group of caregivers ($a_{vs} = 1 \quad \forall v\in\mathcal{V}, s\in\mathcal{S}$). If we further ignore processing times ($p_{is} = 0 \quad \forall i\in\mathcal{C}, s\in\mathcal{S}$) and minimize total distance traveled exclusively ($\lambda_1 = 1; \lambda_2 = 0; \lambda_3 = 0$), the problem results in an mTSPTW. According to Toth and Vigo [14], this problem is $\mathcal{NP}$-hard. Likewise, Bertels and Fahle [3] show that already the HHCRSP without temporal interdependencies is $\mathcal{NP}$-hard as it contains the set-covering problem.

3.3 Example

For illustrating the HHCRSP, we consider a problem with seven patients and a central office, all randomly located in an area of 50×50 distance units. Traveling distances as well as traveling times are assumed to be Euclidean. The qualifications of three HCC employees regarding four types of services are shown in Table 3. The service requirements of the patients are shown together with time windows and time distances $\delta^{\min}$ and $\delta^{\max}$ in Table 4. The set of patients who require a single service is $\mathcal{C}^{s} = \{1, 2, 3, 4, 5\}$, double service patients are $\mathcal{C}^{d} = \{6, 7\}$. A simultaneous double service is required by one patient ($\mathcal{C}^{\text{sim}} = \{6\}$) and a double service with given precedence is required by one other patient ($\mathcal{C}^{\text{prec}} = \{7\}$). Here, the service operations have to be started within a distance of $\delta_7^{\min} = 10$ to $\delta_7^{\max} = 30$ time units. We set processing times $p_{is} = 10$ time units for all service operations. The objective function considers all subgoals at equal weights $\lambda_1 = \lambda_2 = \lambda_3 = \frac{1}{3}$.

The optimal routing to this instance is shown in Fig. 1. The total distance traveled by staff is $D = 306$. The solution is further represented as a time-space diagram in Fig. 2. It can be seen that the time windows are met for all patients except for patient 5, whose service starts with a tardiness of 22 min. Hence, we observe $T = T^{\max} = 22$. It can also be seen that no staff member has to wait at a patient for the reason of arriving too early with respect to the time window. However, waiting times occur at patients 6 and 7 because the staff member who arrives at first at the patient's home has to wait for a colleague to perform the double service. These waiting times appear as horizontal lines in Fig. 2. At patient 6, staff member 3 waits for staff member 2 to start the simultaneous service. Regarding patient 7, staff member 1 waits at the central office and begins the route at time 10 because the service operation has to start at least $\delta_7^{\min} = 10$ time units after the preceding service operation performed by staff member 2.

4 Heuristic solution methods

In this section, we present several heuristic solution methods for solving large instances of the HHCRSP with up to 300 patients. Section 4.1 presents a new solution representation that can accommodate single services and double services. Initial solutions are produced using a construction heuristic described in Section 4.2. In Section 4.3, we propose neighborhood structures for improving initial solutions by a local search. In Section 4.4, we propose an alternative local search procedure where the single and double service neighborhoods are merged, respectively, into one large neighborhood. In Section 4.5, an Adaptive Variable Neighborhood Search strategy is presented which aims at finding solutions beyond the quality of local optimality.

Table 3 Qualifications of staff

Staff member v	Qualifications a_{vs}			
	$s = 1$	$s = 2$	$s = 3$	$s = 4$
1	1	1	0	0
2	0	0	1	1
3	0	0	1	0

Table 4 Service requirements of the patients

Patient i	Service requirements r_{is}				Time window		Time distances	
	1	2	3	4	e_i	l_i	$\delta_i^{\min}$	$\delta_i^{\max}$
1	0	0	1	0	95	148	–	–
2	0	0	1	0	60	127	–	–
3	0	0	1	0	15	45	–	–
4	1	0	0	0	32	91	–	–
5	0	1	0	0	26	90	–	–
6	0	0	1	1	58	114	0	0
7	1	0	0	1	12	50	10	30

4.1 Matrix solution representation

A solution to the HHCRSP can be represented in a matrix scheme $O = (o_{vk})$, where v denotes a staff member and k denotes an enumeration index for the patients. The solution of Fig. 2 is represented within this scheme as is shown in Fig. 3. Rows of the matrix scheme are associated with staff members showing all services he or she is involved in. Columns $k = 1, \ldots, |\mathcal{C}|$ of the scheme indicate the patient,

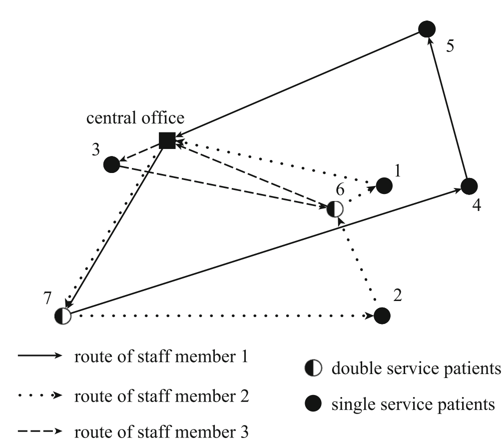

Fig. 1 Optimal routing for the example

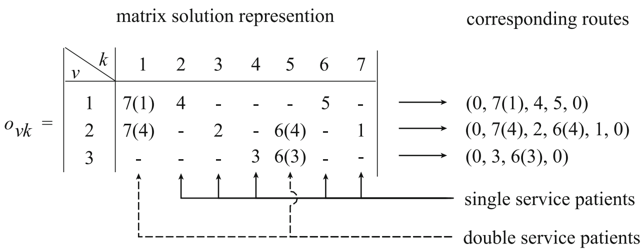

Fig. 3 Example of the matrix representation of a solution

who is considered as the kth candidate in the route construction process described below. In the solution of Fig. 3, patient 7 is the first patient considered. Since this patient requires a double service of types 1 and 4, two qualified staff members (1 and 2) are chosen. The next patient considered is patient 4, who is a single service patient and, thus, the only qualified caregiver $v = 1$ serves him or her. Continuing this way, one or two available staff members are assigned in each column. Every column corresponds to exactly one patient indicating the caregiver(s) who are involved in the required service. Since the rows reflect the projected service orders for the caregivers we can directly read out the corresponding routes from matrix O, see Fig. 3.

This HHCRSP representation scheme has the following properties and advantages:

- It is able to capture all combinations of assignments of employees to patients.
- It reveals the visiting order of the patients for each caregiver.
- It avoids dead-locks which would occur if the operations of simultaneous double services of two patients would be assigned crosswise to two staff members.
- It enables an efficient computation of the corresponding service start times by a straight forward iteration from column to column.
- It simplifies the definition of neighborhoods for local search procedures.

The arrival times of the staff at the patients and the service start times are derived from the matrix representation as follows. Let b_{jv} denote the arrival time of a caregiver v at patient j. For the first patient visited on v's route, b_{jv} is set to the traveling distance from the central office $b_{jv} = d_{0j}$. For all other patients, the arrival time is determined as $b_{jv} = t_{ivs} + p_{is} + d_{ij}$, where i denotes the patient that precedes j in v's route and s refers to the service provided at i. Next, starting times for the services are derived from the arrival times at the patients as follows. For a single service patient $j \in \mathcal{C}^{\mathrm{s}}$, the start time t_{jvs} is determined by the begin of the time window e_j and the arrival time b_{jv} of the assigned staff member v by

$$t_{jvs} = \max\{e_j; b_{jv}\}. \tag{15}$$

For patients $j \in \mathcal{C}^{\mathrm{sim}}$, the start times of the service operations s_1 and s_2 to be executed simultaneously by employees v_1 and v_2 are calculated by

$$t_{jv_1s_1} = t_{jv_2s_2} = \max\{e_j; b_{jv_1}; b_{jv_2}\}. \tag{16}$$

For patients $j \in \mathcal{C}^{\mathrm{prec}}$, the start time of operation s_1 provided by v_1 is determined by (17) like for a single service. The start time of the second service operation s_2 is computed in (18) with respect to the start time of s_1, the minimal time distance $\delta_j^{\min}$ between the two operations, and the arrival time of the staff member v_2 performing s_2.

$$t_{jv_1s_1} = \max\{e_j; b_{jv_1}\} \tag{17}$$

$$t_{jv_2s_2} = \max\left\{t_{jv_1s_1} + \delta_j^{\min}; b_{jv_2}\right\} \tag{18}$$

If it turns out that v_2 arrives too late at the patient, i. e. the maximal time distance $\delta_j^{\max}$ between the services s_1 and s_2 is overshoot $\left(b_{jv_2} > t_{jv_1s_1} + \delta_j^{\max}\right)$, the start time $t_{jv_1s_1}$ of the first operation is postponed by

$$t_{jv_1s_1} = t_{jv_2s_2} - \delta_j^{\max}. \tag{19}$$

Using formulas (15)–(19), the start times of all services are computed iteratively for columns $k = 1, 2, \ldots, |C|$ in the matrix scheme. Table 5 shows the results of this process with respect to the routes in the example. For instance, the

Fig. 2 Time-space diagram of the optimal solution

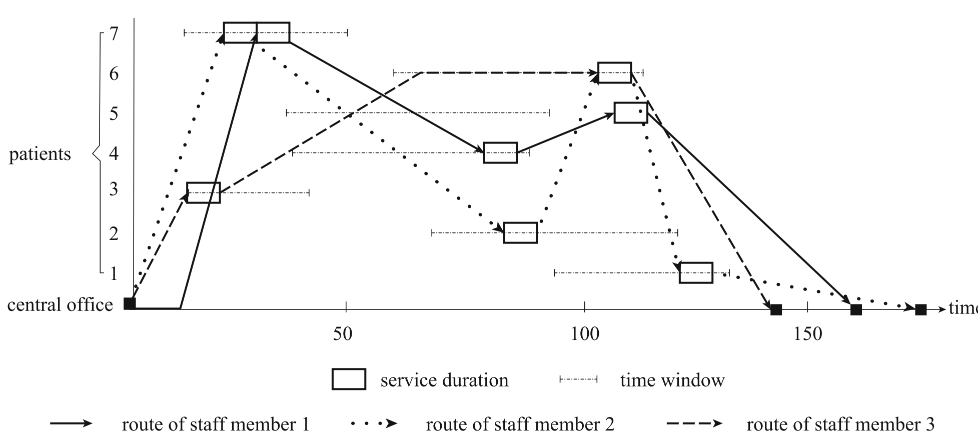

Table 5 Computation of service start times

k	1	2	3	4	5	6	7
$i(s)$	7(1,4)	4(1)	2(3)	3(3)	6(3,4)	5(2)	1(3)
$t_{i,1,s}$	23	84	–	–	–	112	–
$t_{i,2,s}$	33	–	82	–	107	–	124
$t_{i,3,s}$	–	–	–	15	107	–	–
Computed by	(17, 18)	(15)	(15)	(15)	(16)	(15)	(15)

first patient (patient 7) requires a double service with given precedence (service of type 1 before service of type 4). These service operations are assigned to caregivers 1 and 2, who can both arrive at this patient at time $b_{7,1} = b_{7,2} = 23$. Caregiver 1 can start its job immediately because the time window of patient 7 starts at $e_7 = 12$. From (18), caregiver 2 starts the second service at time $t_{7,2,4} = 33$ because of the minimum time distance $\delta_7^{\min} = 10$.

4.2 Generation of initial solutions

A feasible initial routing is constructed by means of the following heuristic. First, the patients are sorted by increasing end of their time window. This gives priority to urgent services in the subsequent iterative construction process. Next, the algorithm assigns each patient to the route of one or two caregivers depending on the service requirements of the considered patient. If the considered patient requires a single service, the algorithm selects the staff member who can arrive at this patient earliest among all suitably qualified employees. If the considered patient requires a double service, those two suitably qualified caregivers are selected that can arrive earliest at the patient. This procedure is repeated until all patients are assigned to routes. The outcome of this procedure is stored in the matrix $O = (o_{vk})$, where column k is produced in the kth iteration of the algorithm. The outline of the heuristic is shown in Fig. 4.

```
INITIAL ROUTING
evaluate sort                              ▷ Sort patients from smallest to largest l_i
for k = 1 → |C| do
    if sort[k] requires a single service then
        s = the type of service required by sort[k]
        select v ∈ V|a_vs = 1 with earliest arrival time at i
        o_vk = sort[k]                     ▷ Assign service to caregiver
    end if
    if sort[k] requires double service then
        s_1 = the type of the first service required by sort[k]
        s_2 = the type of the second service required by sort[k]
        select v̄, v̿ ∈ V with a_v̄s1 = a_v̿s2 = 1 and earliest arrival times at i
        o_v̄k = sort[k](s_1); o_v̿k = sort[k](s_2)   ▷ Assign services to caregivers
    end if
end for
return O
```

Fig. 4 Algorithm for constructing an initial solution

4.3 Neighborhood structures and local search

To improve the quality of initial solutions, we propose a simple local search technique at first. The search is performed on neighborhood structures that are defined within the matrix scheme. In Figs. 5 and 6, we sketch eight neighborhoods, namely intra-shift, intra-swap, inter-shift, and inter-swap moves for both single services and double services. We first describe these moves for single service patients. An intra-shift move shifts a column representing a single service patient to another position within matrix O, where the assignment of the patient to the staff member is retained, see Fig. 5a. An intra-swap move interchanges two columns representing two single service patients that are assigned to the same staff member, see Fig. 5b. An inter-shift move takes a patient out of the route of an employee and inserts the patient into the route of another employee, see Fig. 5c. An inter-swap move interchanges two single service patients that are assigned to two different routes, see Fig. 5d.

For double service patients, we define corresponding neighborhoods by double-intra-shift, double-intra-swap, double-inter-shift, and double-inter-swap moves. The moves are illustrated in Fig. 6a–d. The idea of these neighborhoods is to jointly move both services belonging to one double service patient in order to avoid waiting times of staff members, which could occur if each interdependent service operation would be moved alone. It has to be mentioned that the introduced intra-shift-neighborhoods can produce redundant solutions. This is observed if feasible shifts in the matrix representation lead to the same routes. An example is shown in Fig. 7. Consider the two shifts of column $k = 2$ beyond positions 6 and 7 as pointed out in Fig. 7a. Both shifts lead to different matrices but to the same routing, see Fig. 7b and c. In order to speed up the solution process, we suppress redundant solutions by shifting the considered column not to an arbitrary position, but right behind another column that also contains a patient in the same route. Following this way, the solution of Fig. 7c is circumvented because column 7 does not contain a patient in the same route as column 2.

Initial solutions to the HHCRSP are improved by a steepest descent search, which explores all eight neighborhoods sequentially until no improvement is found.

4.4 Merged neighborhoods

The described local search investigates a total of eight different neighborhoods. Clearly, the solution that is found in the end depends on the sequence in which these neighborhoods are searched. The proposed local search just steps through the neighborhoods in a preset and fixed sequence, e. g. starting with single-intra-shifts that just move single

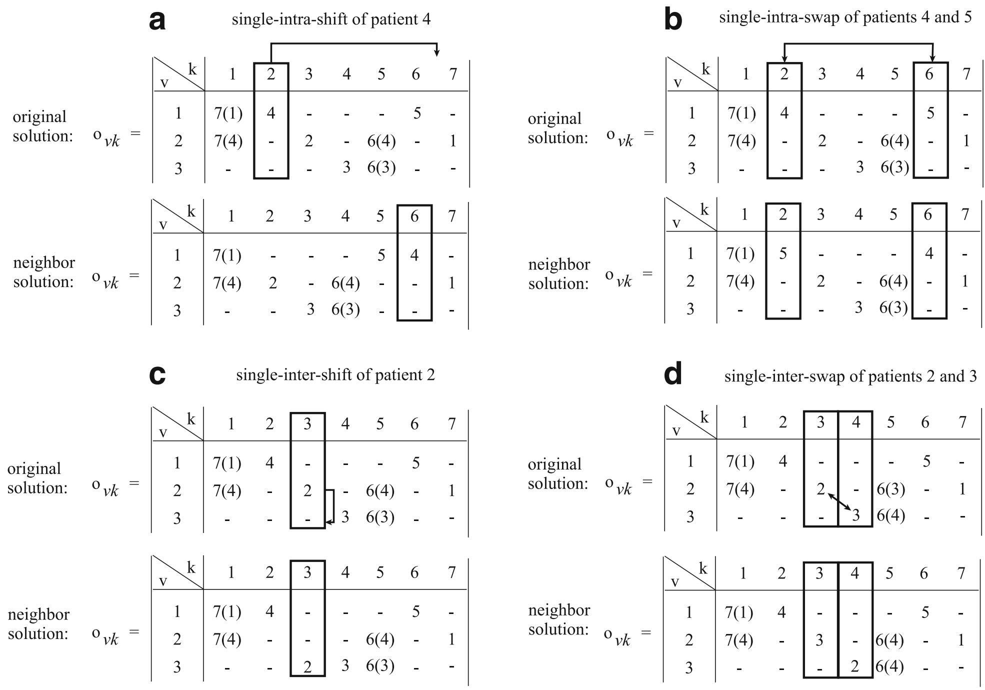

Fig. 5 Examples of single service neighborhood moves

services within their already assigned routes and ending with double-inter-swaps where related services of different routes are exchanged.

In order to diminish the influence of the neighborhood sequence on the solution quality, we merge the four single service neighborhoods into a single large neighborhood. The same is done with the four double service neighborhoods. In doing so, the steepest descent search identifies the best move among all single service neighborhoods in each iteration. If no improvement is found, the search is continued within the merged double service neighborhood. The process is continued until no further improvement is possible.

An alternative for coping with the issue of finding a neighborhood sequence is provided in the next subsection, where an adaptive component within a metaheuristic identifies a promising sequence for a particular problem instance under consideration.

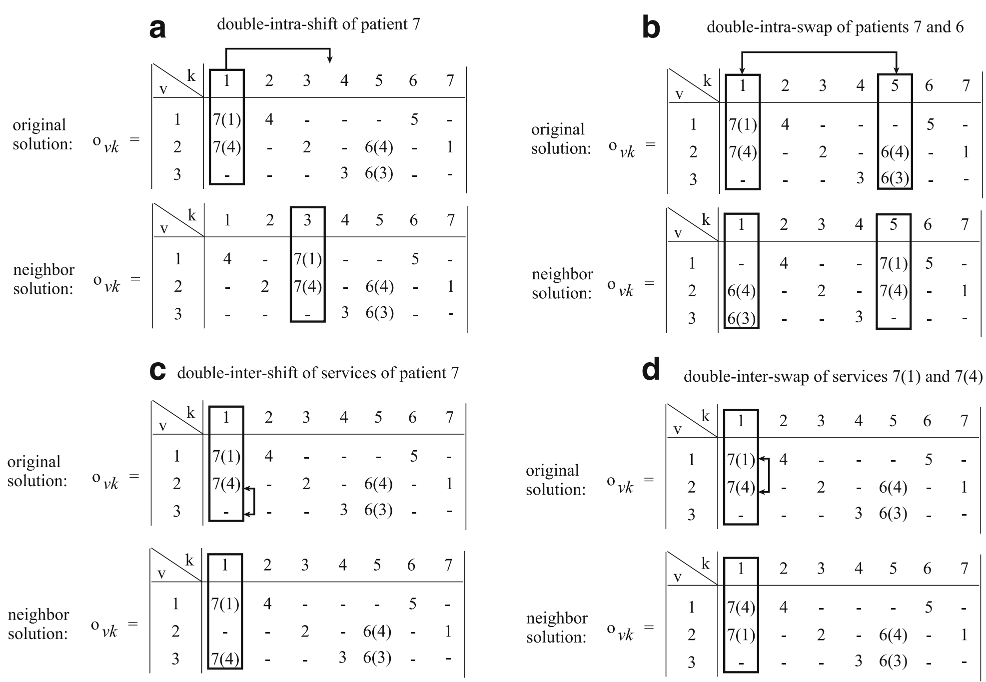

Fig. 6 Examples of double service neighborhood moves

Fig. 7 Redundant neighbors

matrix solution representation | corresponding routes

a original solution: $o_{vk} =$

v \ k	1	2	3	4	5	6	7	corresponding routes
1	7(1)	4	-	-	-	5	-	(0, 7(1), 4, 5, 0)
2	7(4)	-	2	-	6(4)	-	1	(0, 7(4), 2, 6(4), 1, 0)
3	-	-	-	3	6(3)	-	-	(0, 3, 6(3), 0)

b neighbor solution: $o_{vk} =$

v \ k	1	2	3	4	5	6	7	corresponding routes
1	7(1)	-	-	-	5	4	-	(0, 7(1), 5, 4, 0)
2	7(4)	2	-	6(4)	-	-	1	(0, 7(4), 2, 6(4), 1, 0)
3	-	-	3	6(3)	-	-	-	(0, 3, 6(3), 0)

c redundant neighbor: $o_{vk} =$

v \ k	1	2	3	4	5	6	7	corresponding routes
1	7(1)	-	-	-	5	-	4	(0, 7(1), 5, 4, 0)
2	7(4)	2	-	6(4)	-	1	-	(0, 7(4), 2, 6(4), 1, 0)
3	-	-	3	6(3)	-	-	-	(0, 3, 6(3), 0)

4.5 Adaptive variable neighborhood search

In order to avoid getting trapped in poor local optima, Hansen and Mladenović [10] propose to change the neighborhood structure systematically within a local search, which is referred to as Variable Neighborhood Search (VNS). VNS investigates neighborhoods in a preset order. Each time a solution of a better quality is found, the search is restarted with the first neighborhood to intensify search around the new solution.

In this paper, we take up the concept of VNS and extend it to an Adaptive Variable Neighborhood Search (AVNS) metaheuristic for the HHCRSP. The AVNS adapts the sequence in which neighborhoods are searched by VNS to the needs of the problem instance under consideration. This process aims at properly configuring the underlying VNS for the problem instance. For this purpose, we first test the individual progress made by the neighborhoods while performing a certain number of steps in a steepest descent walk. Then, the neighborhoods are ranked by decreasing performance, which lets VNS start with the neighborhood that has the highest potential for delivering good solutions. The procedure terminates if no improvement is found or if a runtime limit is reached. An outline of AVNS is sketched in Fig. 8.

```
AVNS
O_best = current solution                     ▷ best solution known so far
determine rankNH ▷ ranked sequence of neighborhoods derived
from steepest descent walk
l = 1                                         ▷ start with the first neighborhood
while stopping condition is not satisfied do  ▷ perform VNS
    while l ≤ 8 do          ▷ as long as another neighborhood is left
        Ō = LS(rankNH[l])           ▷ best neighbor found within
neighborhood rankNH[l]
        if (Z(Ō) ≤ Z(O_best)) then           ▷ check solution quality
            O_best = Ō                       ▷ accept new best solution
            l = 1          ▷ re-start VNS with the first neighborhood
        else
            l = l + 1 ▷ continue VNS with the next neighborhood
        end if
    end while
end while
return O_best                                 ▷ return best solution
```

Fig. 8 Adaptive variable neighborhood search

5 Computational experiments

In the following, we conduct computational experiments for assessing the problem difficulty of the HHCRSP, for evaluating the performance of the heuristics, and for investigating the suitability of the approach under various problem settings. All algorithms have been implemented in JAVA. The computational experiments are performed on a 3.40 GHz Intel Core computer.

5.1 Generation of test instances

We use seven sets of randomly generated test instances (A to G) that contain ten instances each, see Table 6. The sets differ in the number of patients (from 10 to 300) and the size of the staff (from 3 to 40 employees). Note that planners

Table 6 Characteristics of instance sets

Data set	# instances	$\lvert\mathcal{C}\rvert$	$\lvert\mathcal{C}^s\rvert$	$\lvert\mathcal{C}^d\rvert$	$\lvert\mathcal{V}\rvert$
A	10	10	7	3	3
B	10	25	17	8	5
C	10	50	35	15	10
D	10	75	52	23	15
E	10	100	70	30	20
F	10	200	140	60	30
G	10	300	200	100	40

face instances with up to 200 patients per day in practice (set F), cf. [5]. With set G, we exceed this problem size in order to test our methods under particular challenging planning conditions. In all test instances, 15 % of the patients require a simultaneous double service and 15 % require a double service with precedence. The rest of the patients are single service patients. 30 % double service patients might overestimate the frequency which is observed for such services in practice. However, we use this setting to test our methods thoroughly. The percentage of double services is later subject of a sensitivity analysis in the experiments.

For the double service patients with precedence, a minimal time distance $\delta_i^{\min}$, measured in minutes, is randomly drawn from the interval [0, 60]. A value for $\delta_i^{\max}$ is derived by adding a randomly drawn value from [0, 60] to $\delta_i^{\min}$. The patients and the central office are placed at random locations in the area of 100×100 distance units. Traveling distances d_{ij} $(i, j \in \mathcal{C}^0)$ among the patients' homes and the central office are Euclidean. Six types of services $\mathcal{S} = \{1, \ldots, 6\}$ are considered. The staff members are grouped into two subsets with different qualifications. Each caregiver of the first group is qualified for providing at most three services, which are randomly drawn from the subset $\{1, 2, 3\}$ of $\mathcal{S}$. Accordingly, each caregiver of the second group is qualified for providing at most three services from subset $\{4, 5, 6\}$. It is ensured that each type of service can be provided by at least one caregiver. All processing times p_{is}, measured in minutes, are random integers drawn from the interval [10, 20]. The time windows are of length 120 min (2 h) and are randomly placed within a daily planning period of 10 h. We have no special preference for the three subgoals and, thus, set equal weights $\lambda_1 = \lambda_2 = \lambda_3 = \frac{1}{3}$. Note that traveling distances are equal to the traveling times in the test instances, i. e. the caregivers travel at constant speed of 1 distance unit per time unit. This allows using the same weights for these measures in the objective function. Clearly, in a practical setting, the HCC has to provide actual cost rates for these performance measures. The above setting just serves the purpose of testing the method in an environment where all three objectives are viewed relevant. Different weights are tested in a sensitivity analysis.

5.2 Computational results

5.2.1 Exact solution of the optimization model

In the first test, we attempt to solve the optimization problem exactly by applying the MILP solver ILOG Cplex 12.3. We allow a runtime of 10 h per instance. The computational results are presented in Table 7. The table reports for each instance and solution the lower bound LB, the objective function value Z, the relative gap = (Z−LB)/Z as computed by Cplex (interpreted as the percentage of the current cost Z that is saved if a solution with cost LB actually exists) and the consumed runtime cpu (in seconds). It can be seen that instances of set A are solved to optimality within at most 20 seconds. For set B, the solver finds feasible solutions for all instances but cannot prove optimality within 10 h of runtime. The observed gaps vary strongly between 2 % and 80 %. For sets C to E, Cplex is no longer able to even find feasible solutions, except for just one instance. For these sets, the solver consumes the complete runtime and delivers at least lower bounds for the test instances. In order to also provide lower bounds for sets F and G, the synchronization constraints (11) and (12) are relaxed to reduce the size of the model. For the resulting mTSPTW (with a heterogeneous fleet and compatibility restrictions), Cplex finds lower bounds within the given runtime which, however, are apparently weak. Since the MILP approach allows to solve only very small instances of the HHCRSP, it is concluded that powerful heuristics are needed already for instances with 25 patients and 5 employees (set B).

5.2.2 Performance of proposed heuristics

In the second test, we solve all instances with the proposed heuristics. Initial solutions (Init) are produced as described in Section 4.2. For the local search (LS), we use the neighborhood structures presented in Section 4.3 evaluating all single service neighborhoods at first, followed by the double service neighborhoods. The local search for the merged neighborhoods (MN) is performed as described in Section 4.4. In order to find a suitable sequence for the neighborhoods in the AVNS, five iterations of a steepest descent walk are done with each of the neighborhoods individually. Then, the best performing neighborhood is run in AVNS first, followed by the second best, and so on. We consider two configurations of the metaheuristic, namely AVNS-TL and AVNS. For AVNS-TL, we set a runtime limit that corresponds to the runtime consumed by MN. For AVNS, we allow a longer runtime of 2 h per instance. In this way, we get a deeper understanding of the advantages of metaheuristic search with respect to better solution quality and its disadvantage regarding longer runtimes.

Table 7 shows the cost Z of the obtained solution for each instance and each of the heuristics. Runtimes are reported for LS, MN, AVNS-TL and AVNS only, because initial solutions are always produced within less than a second even for the largest instances. For AVNS, we additionally report the gap to the lower bound reported by Cplex. Average results for the instance sets are reported in Table 8. We observe that the best heuristic solutions for set A are achieved by AVNS(-TL). The metaheuristic solves 8 out of 10 instances to optimality and returns near optimal solutions for the rest. Its runtime is below one second per instance. For set B, five of the best known solutions are found by

Table 7 Numerical results for all instances

Instance	LB	Cplex			Init	LS		MN		AVNS-TL		AVNS		
		Z	gap	cpu	Z	Z	cpu	Z	cpu	Z	cpu	Z	gap	cpu
A_1	218.2	**218.2**[a]	0.0 %	2	464.4	**218.2**	< 1	**218.2**	< 1	**218.2**	< 1	**218.2**	0.0 %	< 1
A_2	246.6	**246.6**	0.0 %	5	1034.4	248.1	< 1	248.1	< 1	248.1	< 1	248.1	0.6 %	< 1
A_3	305.8	**305.9**	0.0 %	7	569.1	**305.9**	< 1	316.4	< 1	316.4	< 1	**305.9**	0.0 %	< 1
A_4	186.9	**186.9**	0.0 %	8	676.3	**186.9**	< 1	**186.9**	< 1	**186.9**	< 1	**186.9**	0.0 %	< 1
A_5	189.5	**189.5**	0.0 %	2	421.2	201.1	< 1	192.0	< 1	192.0	< 1	192.0	1.3 %	< 1
A_6	200.1	**200.1**	0.0 %	2	1282.3	**200.1**	< 1	**200.1**	< 1	**200.1**	< 1	**200.1**	0.0 %	< 1
A_7	225.4	**225.4**	0.0 %	1	933.6	**225.4**	< 1	**225.4**	< 1	**225.4**	< 1	**225.4**	0.0 %	< 1
A_8	232.0	**232.0**	0.0 %	4	296.0	**232.0**	< 1	**232.0**	< 1	**232.0**	< 1	**232.0**	0.0 %	< 1
A_9	222.3	**222.3**	0.0 %	20	781.2	230.6	< 1	230.6	< 1	**222.3**	< 1	**222.3**	0.0 %	< 1
A_{10}	225.0	**225.0**	0.0 %	1	1540.4	**225.0**	< 1	**225.0**	< 1	**225.0**	< 1	**225.0**	0.0 %	< 1
B_1	378.7	1134.9	66.6 %	36000	2215.0	491.3	< 1	490.3	< 1	**458.9**	< 1	**458.9**	17.5 %	< 1
B_2	427.9	**476.2**	10.1 %	36000	1228.7	580.9	< 1	544.2	< 1	580.9	< 1	580.9	26.3 %	< 1
B_3	391.2	**399.2**	2.0 %	36000	3663.8	442.0	< 1	461.6	< 1	431.4	< 1	431.4	9.3 %	< 1
B_4	330.4	**576.0**	42.6 %	36000	1402.4	587.3	< 1	577.5	< 1	587.3	< 1	587.3	43.7 %	< 1
B_5	311.0	599.4	48.1 %	36000	2469.3	431.5	< 1	**396.9**	< 1	391.1	< 1	391.1	20.5 %	< 1
B_6	274.2	1357.5	79.8 %	36000	1058.5	545.9	< 1	**534.7**	< 1	545.9	< 1	545.9	49.8 %	< 1
B_7	310.6	432.3	28.2 %	36000	1184.3	366.2	< 1	**355.5**	< 1	356.6	< 1	356.6	12.9 %	< 1
B_8	332.4	**357.8**	7.1 %	36000	504.2	421.6	< 1	425.3	< 1	410.9	< 1	410.9	19.1 %	< 1
B_9	293.7	**403.8**	27.3 %	36000	782.7	591.2	< 1	532.5	< 1	487.9	< 1	487.9	39.8 %	< 1
B_{10}	381.0	629.9	39.5 %	36000	3216.6	507.6	< 1	**500.4**	< 1	**500.4**	< 1	**500.4**	23.9 %	< 1
C_1	401.1	–	–	36000	2263.7	1246.7	< 1	1259.9	< 1	**1123.6**	< 1	**1123.6**	64.3 %	< 1
C_2	314.9	–	–	36000	1063.3	749.2	< 1	**673.8**	< 1	677.0	< 1	677.0	53.5 %	< 1
C_3	323.5	–	–	36000	1359.5	693.1	< 1	665.7	< 1	**642.4**	< 1	**642.4**	49.6 %	< 1
C_4	329.4	–	–	36000	1460.1	594.4	< 1	581.9	< 1	**580.4**	< 1	**580.4**	43.2 %	< 1
C_5	404.2	–	–	36000	1412.3	779.4	< 1	775.7	< 1	**754.6**	< 1	**754.6**	46.4 %	< 1
C_6	308.2	–	–	36000	2313.0	993.6	< 1	994.8	< 1	**951.6**	< 1	**951.6**	67.6 %	< 1
C_7	315.7	4753.5	93.4 %	36000	772.0	654.1	< 1	633.8	< 1	**577.4**	< 1	**577.4**	45.3 %	< 1
C_8	336.3	–	–	36000	949.2	566.4	< 1	566.2	< 1	**540.6**	< 1	**540.6**	37.8 %	< 1
C_9	306.5	–	–	36000	1072.6	708.7	< 1	638.6	< 1	**608.7**	< 1	**608.7**	49.6 %	< 1
C_{10}	386.4	–	–	36000	1367.5	749.4	< 1	728.6	< 1	**679.3**	< 1	**679.3**	43.1 %	< 1
D_1	456.7	–	–	36000	4072.6	1480.8	1	1657.3	5	**1321.8**	5	**1321.8**	65.4 %	5
D_2	336.8	–	–	36000	1361.7	931.6	1	898.3	4	**892.7**	4	**892.7**	62.3 %	4
D_3	355.7	–	–	36000	1482.0	891.6	1	883.3	4	**819.4**	4	**819.4**	56.6 %	6
D_4	379.9	–	–	36000	2780.8	964.3	1	899.8	4	**877.4**	4	**877.4**	56.7 %	4
D_5	372.2	–	–	36000	2038.9	1055.1	1	1008.8	5	**872.1**	5	**872.1**	57.3 %	7
D_6	368.8	–	–	36000	1984.9	980.3	1	942.5	5	**835.2**	5	**835.2**	55.8 %	8
D_7	333.4	–	–	36000	1025.8	717.5	2	715.4	6	**706.3**	6	**706.3**	52.8 %	7
D_8	373.3	–	–	36000	1177.6	854.7	2	833.5	4	**811.4**	4	**811.4**	54.0 %	7
D_9	362.4	–	–	36000	1325.0	869.5	1	842.7	6	**860.3**	6	**860.3**	57.9 %	7
D_{10}	434.4	–	–	36000	6591.8	1531.5	1	1396.5	3	**1306.6**	3	**1306.6**	66.8 %	4
E_1	406.0	–	–	36000	4254.5	1708.0	10	1690.3	17	**1604.9**	17	**1604.9**	74.7 %	31
E_2	411.9	–	–	36000	1682.9	1101.9	10	**1153.6**	19	1159.7	14	1159.7	64.5 %	27
E_3	437.2	–	–	36000	1651.4	1035.1	8	1031.9	22	**986.4**	14	**986.4**	55.7 %	28
E_4	384.3	–	–	36000	1475.9	1059.4	10	1015.0	22	**871.0**	19	**871.0**	55.9 %	37
E_5	392.4	–	–	36000	1627.0	1108.3	8	1048.0	22	**1018.0**	19	**1018.0**	61.5 %	29
E_6	390.2	–	–	36000	1669.5	1089.8	8	1073.1	22	**1003.0**	19	**1003.0**	61.1 %	36
E_7	374.4	–	–	36000	1296.3	994.3	8	1001.4	21	**921.1**	20	**921.1**	59.4 %	38

Table 7 (continued)

Instance	LB	Cplex			Init	LS		MN		AVNS-TL		AVNS		
		Z	gap	cpu	Z	Z	cpu	Z	cpu	Z	cpu	Z	gap	cpu
E_8	407.7	–	–	36000	1465.9	996.9	6	957.9	20	**884.6**	19	**884.6**	53.9 %	36
E_9	422.1	–	–	36000	1792.6	1226.7	8	1236.0	19	**1131.7**	18	**1131.7**	62.7 %	34
E_{10}	419.1	–	–	36000	1600.1	1053.6	11	1077.6	19	**1066.6**	16	**1066.6**	60.7 %	31
F_1	445.1	–	–	36000	2845.0	1837.4	210	1780.9	414	1790.2	414	**1721.4**	74.1 %	889
F_2	457.9	–	–	36000	2667.2	1888.3	210	1848.9	419	1821.8	419	**1763.8**	74.0 %	909
F_3	481.8	–	–	36000	2545.0	1700.4	171	1652.3	396	1659.2	396	**1549.6**	68.9 %	868
F_4	417.4	–	–	36000	2360.6	1504.0	244	1451.3	608	1486.8	608	**1420.4**	70.6 %	1321
F_5	452.3	–	–	36000	2659.6	1825.0	288	1772.9	478	1819.2	478	**1701.9**	73.4 %	1145
F_6	367.1	–	–	36000	2655.2	1834.3	215	1745.6	413	1681.0	413	**1639.7**	77.6 %	836
F_7	408.1	–	–	36000	2322.6	1533.3	270	1529.5	504	1525.3	504	**1384.3**	70.5 %	1294
F_8	454.3	–	–	36000	2396.8	1749.5	195	1659.1	413	1611.3	413	**1544.6**	70.6 %	924
F_9	426.8	–	–	36000	2725.5	1926.1	346	1784.6	556	1856.1	556	**1572.9**	72.9 %	1642
F_{10}	441.8	–	–	36000	2790.2	1906.2	237	1798.2	450	1830.8	450	**1581.0**	72.1 %	1326
G_1	455.0	–	–	36000	3677.6	2414.9	1458	2317.0	3078	2452.2	3078	**2248.0**	79.8 %	7200
G_2	463.1	–	–	36000	3541.3	2478.5	1472	2395.4	3639	2404.7	3639	**2316.1**	80.0 %	7200
G_3	464.3	–	–	36000	3201.2	2077.4	1402	1988.9	2647	2062.4	2647	**1885.3**	75.4 %	7147
G_4	461.5	–	–	36000	3200.6	2140.5	2014	2062.3	2992	2215.1	2992	**2023.2**	77.2 %	7200
G_5	449.4	–	–	36000	3860.0	2533.4	1249	2420.4	2796	2494.3	2796	**2247.6**	80.0 %	7200
G_6	471.5	–	–	36000	3449.6	2313.7	1580	2277.6	2861	2386.5	2861	**2144.4**	78.0 %	7200
G_7	459.1	–	–	36000	3205.8	2142.6	1402	2061.2	3085	2072.2	3085	**1971.5**	76.7 %	6934
G_8	472.4	–	–	36000	3290.5	2198.3	1354	2046.3	2819	2174.4	2819	**1987.4**	76.2 %	7200
G_9	473.3	–	–	36000	3921.2	2686.5	1500	2585.7	3095	2540.6	3095	**2415.5**	80.4 %	7023
G_{10}	451.6	–	–	36000	3759.7	2430.2	1224	2411.7	2581	2587.0	2581	**2373.4**	81.0 %	7003

[a]Bold values indicate best known solution for an instance (For set A, these values indicate the optimal solutions)

Cplex, four best solutions are obtained by MN, and two by AVNS(-TL). This shows that merging the neighborhoods represents a useful solution strategy for these instances. Nevertheless, the lowest average cost for set B is achieved by AVNS(-TL), see Table 8. Although a gap of 26.3 % remains to the lower bounds, the heuristic saves 25 % of cost compared to Cplex's solutions in less than a second of runtime. For instances in set C to G, AVNS is the best performing method as it delivers 48 out of 50 best known solutions. It achieves substantial improvements over the initially constructed solutions and the local optima, delivering significant cost savings even for instances with up to 300 patients and 40 staff members. It is also noticed that the runtimes grow quickly but appear to be still acceptable for a

Table 8 Average numerical results for all instance sets

Set	LB	Cplex			Init	LS		MN		AVNS-TL		AVNS		
		Z	gap	cpu	Z	Z	cpu	Z	cpu	Z	cpu	Z	gap	cpu
A	225.2	**225.2**	0.0 %	5	799.9	227.3	< 1	227.5	< 1	225.6	< 1	225.6	0.2 %	< 1
B	343.1	636.7	35.1 %	36000	1772.6	496.6	< 1	481.9	< 1	**475.1**	< 1	**475.1**	26.3 %	< 1
C	342.6	4753.5	93.4 %	36000	1403.3	773.5	< 1	751.9	< 1	**713.6**	< 1	**713.6**	50.1 %	< 1
D	377.4	–	–	36000	2384.1	1027.7	1	1007.8	5	**930.3**	5	**930.3**	58.6 %	6
E	404.5	–	–	36000	1851.6	1137.4	9	1128.5	19	**1064.7**	19	**1064.7**	61.0 %	33
F	435.3	–	–	36000	2596.8	1770.5	239	1702.3	465	1708.2	465	**1588.0**	72.5 %	1115
G	462.1	–	–	36000	3510.8	2341.6	1466	2256.7	2959	2336.2	2959	**2161.2**	78.5 %	7133

Table 9 Numerical results for different objectives

Set	D pursued				T pursued						$T^{\max}$ pursued			
	Cplex		AVNS		Cplex			AVNS			Cplex		AVNS	
	D	cpu	D	cpu	T	∅p.p.	cpu	T	∅p.p.	cpu	$T^{\max}$	cpu	$T^{\max}$	cpu
A	537	18	544	< 1	16	2	4	16	2	<1	12	4	12	< 1
B	912	7200	938	< 1	744	30	6826	100	4	<1	78	6391	45	< 1

practical application. The runtimes up to set E can be further restricted without deteriorating the solution quality, see column AVNS-TL. However, for the very large instances in sets F and G, AVNS-TL performs worse than MN because the metaheuristic spends a part of its given runtime on finding a suitable neighborhood sequence. Here, a longer runtime must be allowed to obtain the best solutions reported in column AVNS.

5.2.3 Sensitivity analysis

In the previous tests, the three performance measures were equally weighted within the objective function. To investigate the impact of the conflicting goals on the solutions, we treat the performance measures D, T, and $T^{\max}$ separately by minimizing

- the total distance traveled by employees, i. e. $\lambda_1 = 1$, $\lambda_2 = \lambda_3 = 0$,
- the total tardiness of services, i. e. $\lambda_2 = 1$, $\lambda_1 = \lambda_3 = 0$, and
- the maximum tardiness among all patients, i. e. $\lambda_3 = 1$, $\lambda_1 = \lambda_2 = 0$.

To get (if possible) the optimal value of the performance measures, sets A and B are solved again using the MILP solver with a runtime limit of 2 h per instance. Accordingly, sets A and B are solved by the AVNS heuristic under all performance measures. The results are shown in Table 9. Like in the first test, Cplex obtains optimal solutions for set A, whereas for set B merely feasible solutions are obtained. To the contrary, AVNS solves all instances within less than one second under every setting.

When minimizing D, we observe that the solutions of AVNS differ by less than 3 % from the optimal solutions of both sets A and B. This indicates a very good solution quality for AVNS. In other words, the heuristic performs well if we strive for minimizing the traveling costs alone.

When minimizing T, AVNS finds even optimal results for set A. The total tardiness is 16 min for these solutions, which corresponds to an average tardiness of less than 2 min per patient (see column ∅p.p. in Table 9). For the larger instances of set B, the average tardiness per patient grows to 4 min, which is still tolerable in practical applications. At the same time, the solver fails completely in delivering solutions of acceptable service quality as the patients have to face an average of 30 min of waiting time.

Similar results are observed when minimizing $T^{\max}$. Again, AVNS delivers optimal solutions for set A. For set B, AVNS clearly outperforms the solver as it delivers solutions with a maximal tardiness of 45 min whereas the solver produces a maximal tardiness of 1 h and 18 min.

It is concluded that pursuing each of the objectives exclusively is adequately supported by our heuristic. AVNS delivers solutions of very good service quality and very low traveling cost within negligible runtime which makes its application attractive for HCCs.

Finally, we conduct a sensitivity analysis with respect to the number of double service patients contained in

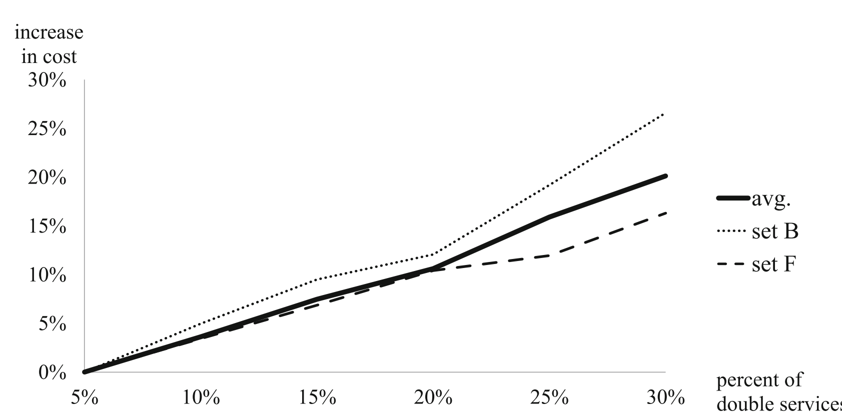

Fig. 9 Sensitivity on the percentage of double services

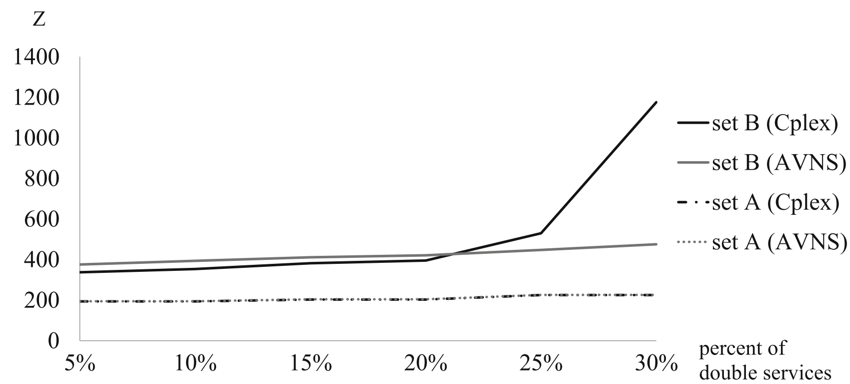

Fig. 10 Cplex vs. AVNS for varied double services

the instances. For this experiment, we produce instances with 5 %, 10 %, . . ., 30 % of double services by transforming some double service patients of the original instances into single service patients. This is done by eliminating one of the services of these double service patients. We apply AVNS to each of these instance sets and identify the change in cost that comes along with different percentages of double service patients. Figure 9 shows the increase in cost (in %) effected by the variation of double services. Here, the cost observed under 5 % of double service patients serves as reference. The figure illustrates the average cost increase over all seven instance sets and, exemplarily, for sets B and F. We observe that the average cost grows almost proportionally to the number of double service patients. This shows that AVNS works reliably, in particular with respect to the synchronization of working plans.

Figure 10 shows the average cost of solutions to sets A and B obtained by Cplex and AVNS under the varied percentages of double services. We see that both methods deliver the same solution quality for any number of double services in set A. Regarding set B, Cplex has a slight advantage for low percentages of double services but it is clearly outperformed by AVNS for the higher percentage values. Note also that Cplex again consumes the complete allowed runtime whereas AVNS returns solutions within just a second. Furthermore, since Cplex does not solve instances of sets C to G, we conclude from our tests that AVNS is clearly preferable as it solves much larger instances, requires much shorter runtimes, and copes well with high numbers of double services.

6 Concluding remarks

A model and solution methods are proposed for the home health care routing and scheduling problem with interdependent services. Services that must be performed by two caregivers simultaneously or within a certain time distance are vitally important in the home care business. However, incorporating them in the planning process of a home care company is hardly supported by the recent planning approaches. We have developed a new heuristic that is capable of handling such synchronization requirements by relying on a sophisticated solution representation scheme. The solutions gained in the experiments indicate that the method consistently achieves low traveling cost for caregivers, low average waiting times for patients, a fair distribution of inevitable tardiness, or a combination thereof. At the same time, very large problem instances with several hundred patients can be treated in acceptable runtime.

Acknowledgments This research is funded by the German Research Foundation (DFG) under reference B02110263.

References

1. Akjiratikarl C, Yenradee P, Drake PR (2007) PSO-based algorithm for home care worker scheduling in the UK. Comput Ind Eng 53:559–583
2. Begur SV, Miller DM, Weaver JR (1997) An integrated spatial DSS for scheduling and routing home-health-care nurses. Interfaces 27(4):35–48
3. Bertels S, Fahle T (2006) A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem. Comput Oper Res 33:2866–2890
4. BMG (2012) Figures and facts: long-term care insurance (04/12) (in German). URL http://www.bmg.bund.de/pflege/zahlen-und-fakten-zur-pflegeversicherung.html, Access 12.11.2012
5. Boß B (2012) Personal communication with head of nibola GbR (an HCC software supplier). URL http://nibola.com/
6. Bräysy O, Dullaert W, Nakari P (2007) Municipal routing problems: a challenge for researchers and policy makers? In: Witlox FJA, Ruijrok CJ (eds) Bijdragen Vervoerslogistieke Werkdagen 2007. Nautilus Academic Books, Zelzate, pp 330–347
7. Bredström D, Rönnqvist M (2008) Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. Eur J Oper Res 191:19–31
8. Cheng E, Rich JL (1998) A home health care routing and scheduling problem. Tech. Rep. TR98-04, Rice University, USA
9. Eveborn P, Flisberg P, Rönnqvist M (2006) LAPS CARE—an operational system for staff planning of home care. Eur J Oper Res 171:962–976

10. Hansen P, Mladenović N (2001) Variable neighborhood search: principles and applications. Eur J Oper Res 130:449–467
11. Kergosien Y, Lenté C, Billaut JC (2009) Home health care problem: an extended multiple traveling salesman problem. In: Multidisciplinary international conference on scheduling: theory and applications (MISTA 2009), pp 85–92
12. Rasmussen MS, Justesen T, Dohn A, Larsen J (2012) The home care crew scheduling problem: preference-based visit clustering and temporal dependencies. Eur J Oper Res 219:598–610
13. Tarricone R, Tsouros AD (eds) (2008) Home care in Europe: the solid facts. Regional Office for Europe of the World Health Organization, Copenhagen
14. Toth P, Vigo D (eds) (2002) The vehicle routing problem. SIAM monographs on discrete mathematics and applications. SIAM, Philadelphia
15. Trautsamwieser A, Hirsch P (2011) Optimization of daily scheduling for home health care services. J Appl Oper Res 3(3):124–136