Flex Serv Manuf J (2016) 28:495–525
DOI 10.1007/s10696-015-9227-1

Daily scheduling of home health care services using time-dependent public transport

Klaus-Dieter Rest[1] · **Patrick Hirsch**[1]

Published online: 13 October 2015

Abstract This paper presents a real-world optimization problem in home health care that is solved on a daily basis. It can be described as follows: care staff members with different qualification levels have to visit certain clients at least once per day. Assignment constraints and hard time windows at the clients have to be observed. The staff members have a maximum working time and their workday can be separated into two shifts. A mandatory break that can also be partitioned needs to be scheduled if the consecutive working time exceeds a certain threshold. The objective is to minimize the total travel- and waiting times of the care staff. Additionally, factors influencing the satisfaction of the clients or the care staff are considered. Most of the care staff members from the Austrian Red Cross (ARC) in Vienna use a combination of public transport modes (bus, tram, train, and metro) and walking. We present a novel model formulation for this problem, followed by an efficient exact solution approach to compute the time-dependent travel times out of the timetables from public transport service providers on a minute-basis. These travel time matrices are then used as input for three Tabu Search based solution methods for the scheduling problem. Extensive numerical studies with real-world data from the ARC show that the current planning can be improved significantly when these methods are applied.

Keywords Home health care · Multimodal routing · Time-dependent travel times · Tabu search

✉ Klaus-Dieter Rest
klausdieter.rest@boku.ac.at

Patrick Hirsch
patrick.hirsch@boku.ac.at

[1] Institute of Production and Logistics, University of Natural Resources and Life Sciences, Vienna, Feistmantelstrasse 4, 1180 Vienna, Austria

1 Introduction

Home health care (HHC) services are ranging from assistance in leading the household and maintenance of social contacts to qualified home nursing. They allow old and frail people a self-determined living in their familiar environment as long as possible. Demographic (e.g., rising life expectancy) and social (e.g., more childless families and prolonged employment) changes are leading to more care dependent people and to a reduced potential of family care. As a result, many industrialized countries are facing a significant increase in demand for HHC services and further rises must be expected. Additionally, HHC service providers are exposed to high cost pressure and are thus, highly interested in suitable decision support systems (DSS). Due to various constraints the planning of the care staff's schedules is quite complex. Matta et al. (2014) depict the main processes and management decisions after analyzing several HHC service providers in France and Italy. In Austria, the planning process is usually still done manually. Hence, DSSs lead presumably not only to better solutions, but also to a reduced planning effort.

The daily HHC scheduling problem can be summarized as follows: A certain number of care staff members with different qualification levels has to visit a set of clients at least once during a day. Assignment constraints and hard time windows of these visits as well as maximum working times per day and mandatory break times have to be observed. The mandatory break times can also be splitted. The objective is to minimize the total travel- and waiting times of the care staff. Additionally, factors influencing the satisfaction of the clients or the staff members are either considered within the objective function or through hard constraints. A detailed formal explanation of the problem can be found in Sect. 3.

The presented research is based on a collaborative project with the Austrian Red Cross (ARC), one of the leading HHC service providers in Austria. According to them, the majority of care staff in urban areas uses public transport (bus, tram, subway, and train) to visit their clients, if there is a well-developed public transport system. In the case of Vienna, more than 90 % of the ARC care staff uses exclusively public transport. As public transport operates on timetables with varying intervals and travel times during the day, time-dependent travel times are considered to model real travel times the best possible way. Public transport also implies multi-modality as different lines of the public transport system could be necessary to reach the homes of the clients (e.g., the combination bus-subway-tram). Existing research in the field of HHC usually considers just the use of a single mode of transport for the staff and to the best of our knowledge, also none of these approaches are based on time-dependent travel times. From a modeling point of view, the presented problem can be seen as a time-dependent vehicle routing problem (TDVRP) with additional HHC-related constraints.

The main contribution of this paper is to present solution approaches to solve the introduced real-world HHC problem with time-dependencies and multi-modality. In a first step we develop an efficient exact solution approach to compute time-dependent travel times out of the timetables from the public transport service providers. Time intervals of 1 min are used. The generated travel time matrices are then used as input for metaheuristic solution approaches. We have designed and implemented three Tabu Search (TS) approaches and compare their solutions against each other as well as with the actual planning of the ARC. The numerical studies are based on real-world data and deal with problem sizes of up to 46 care staff members and 202 visits. The results show that the solution approaches are capable to provide solutions of good quality in short computation times. The presented methods are tailored to the requirements of HHC in urban areas. Nevertheless, it is worth mentioning that the introduced ideas can also be used with adapted constraints in other application areas like for example service technician routing.

Our research on the time-dependent HHC problem differs from the published work by combining both research areas. Thus, we link a complex real-world problem in the field of HHC with the complexity of time-dependent routing. The contributions to the individual areas are as follows. For optimization of daily HHC services, we extend the published work by considering additional constraints related to working time restrictions (e.g., split breaks, minimal working times). In the area of time-dependent routing, we present an efficient way to compute time-dependent travel times out of timetable data as well as an approach to determine the optimal start time of time-dependent routes.

The organization of the paper is as follows: Sect. 2 contains a literature review on HHC as well as on time-dependent routing. In Sect. 3 a problem description is given, including a linear programming formulation. Section 4 explains the modeling aspects of the time-dependency, followed by our TS based solution approaches in Sect. 5. To show the applicability of our approaches, numerical studies with real-world data from the region of Vienna are presented in Sect. 6. Finally, conclusions and an outlook on future research are given in Sect. 7.

2 Literature review

In the following we present an overview on literature about HHC scheduling and time-dependent routing.

2.1 Home health care

The problem we are dealing with is usually addressed in literature as HHC problem. Optimization in the field of HHC is a rather young but quickly evolving research area, due to the rising importance of HHC for today's society. Thus, many different approaches have been published in the recent years. Besides the varying legal and organizational requirements the points of focus are diverse: The first paper in this field we are aware of, comes from Begur et al. (1997). The authors present a sequential savings algorithm to solve a HHC problem in the USA. Cheng and Rich (1998) introduce a model formulation for the HHC problem. They give both, a two- and a three-index formulation and solve them with a construction heuristic as well as with CPLEX as a MIP solver. Bertels and Fahle (2006) list several factors that are relevant for an allocation of clients to care staff members. For solving their HHC problem they have developed a software called PARPAP, a combination of Constraint Programming (CP) with the metaheuristics Simulated Annealing and TS, which is able to solve instances with up to 50 staff members and 326 jobs. Eveborn et al. (2006) developed a DSS called Laps Care for Swedish HHC service providers. They treated the problem as a set partitioning problem and use a solution approach based on repeated matching. Considered modes of transport are car, bicycle, and walking. Nevertheless, these modes of transport are not combined during the trip of a staff member. They present results for real-world instances, including an urban region, where all journeys are made by foot due to the high density of clients. Their recent work on this topic has been published in Eveborn et al. (2009). Akjiratikarl et al. (2007) compare solutions obtained by a Particle Swarm Optimization based metaheuristic with schedules from a local government authority in UK using its existing manual processes and with a proprietary routing software. Elbenani et al. (2008) present a TS algorithm to solve a real-world HHC problem in Canada. A unique feature of this work is that nurses have to collect blood samples from some clients and deliver them to the hospital within a given time window. To facilitate continuity of care for clients requiring several visits a day, penalty costs are introduced if the follow-ups are carried out by different nurses. Bräysy et al. (2009) present some case studies, which aim to optimize several communal services (HHC, transportation of the elderly, and home meal delivery) in Finland by using a commercial VRP solver. For HHC, the authors claim that care staff members can use different modes of transport (walking, car, bicycle, and bus); however, a single average travel speed over all modes is taken. Dohn et al. (2008) use a Branch-and-Price approach to solve real-world instances with up to 150 jobs and 15 staff members. Bredström and Rönnqvist (2007), Bredström and Rönnqvist (2008), Rasmussen et al. (2012), and Mankowska et al. (2013) focused their work on interdependent services, taking into account pairwise synchronization as well as pairwise temporal precedence between jobs. Like Dohn et al. (2008), Bredström and Rönnqvist (2007) and Rasmussen et al. (2012) also present a Branch-and-Price algorithm. To reduce the computational effort, Rasmussen et al. (2012) further analyze different clustering schemes. Bredström and Rönnqvist (2008) and Mankowska et al. (2013) use metaheuristic solution approaches to achieve this goal. Trautsamwieser et al. (2011) present a model formulation for HHC in Austria considering many legal constraints as well as the satisfaction of clients and care staff members through several factors within a weighted objective function. Mandatory breaks are also included if the working time of a staff member exceeds a certain amount of time, however, breaks are always of fixed length and can not be partitioned into smaller parts. Trautsamwieser et al. (2011) also analyze the effects of some natural disasters on the planning. While the focus of their work is on rural areas, Rest et al. (2012) review trends and risks in HHC with a focus on urban regions. Another recent work with a focus on urban regions has been published by Hiermann et al. (2013). The authors present a two-stage approach that uses CP to generate a feasible initial solution that is further improved with one of four implemented metaheuristics (Variable Neighborhood Search, Memetic Algorithm, Scatter Search, and Simulated Annealing). Care staff members are able to use either public transport or cars. Their travel time data is based on estimates from a public transport service provider as well as on floating car data for the road network. However, a single estimate is used for each mode and thus, they do not rely on the actual time of departure. Real-world instances of an HHC service provider in Vienna are used to evaluate their solution approach.

In addition to the daily scheduling of HHC services, there is also an increasing research interest in the periodic/mid-term HHC problem. This problem is characterized by a prolonged planning horizon of a week or more, which induces additional constraints (e.g., rest periods, visiting frequencies, continuity of care) and requires different modeling and solution approaches. Therefore, those who are interested in this type of HHC problem are referred to the recent publications of Cappanera and Scutellà (2014), Liu et al. (2014), or Trautsamwieser and Hirsch (2014) as these already contain comprehensive literature reviews.

2.2 Time-dependent routing

Using time-dependent travel times leads to schedules that are more viable, especially in urban regions. Because of this main benefit, there were still some early efforts to incorporate varying travel times into models and algorithms. Nevertheless, not all of them satisfy the First-In First Out (FIFO) principle, although it is a natural assumption. FIFO, also referred to as non-passing property, implies that a vehicle departing from node i to node j using arc (i, j) at time t will always arrive before or at the same time as a vehicle using the same arc but departing at a later time: $A_{ij}(t_0) \leq A_{ij}(t_1)$ as long as $t_1 > t_0$. Hence, the arrival time function A_{ij} will be a non-decreasing function of time. Malandraki and Daskin (1992) were among the first who stated a mixed integer linear programming formulation of the TDVRP. A stepwise travel time function is assumed, but as waiting at nodes is permitted, the actual travel time function (after computing the shortest paths) behaves like a continuous function, as long as the travel time is not increasing. FIFO is achieved because in times of decreasing travel times, travel time will be substituted with waiting time. However, one will arrive at the same time which does not represent reality. To solve the model formulation, a nearest neighbor and a cutting plane heuristic are presented. Hill and Benton (1992) present a modeling approach, where the time-dependent information is assigned to the nodes instead of the edges. This can be interpreted as the average speed for the area around a node. In this way they wanted to simplify data collection and to reduce the computational effort. Their usage of different speed levels for discrete time periods leads to piecewise constant speed functions. As a result, the travel time function is also piecewise constant and does not satisfy the FIFO property. To validate their approach the authors are mentioning the implementation of a greedy heuristic for a TDVRP in a city with 210 locations and time-dependent travel speeds for 96 time periods. Ichoua et al. (2003) also use piecewise constant speed, but in contrary to Hill and Benton (1992), they do not assume constant speed over the entire length of the link. The travel speed is adjusted when the vehicle crosses the boundaries of the time periods. Hence, the travel time is computed by summing up the time used for each part of the link. This approach results in a travel time function, that is piecewise continuous over the time and thus satisfies FIFO. Fleischmann et al. (2004) provide a modeling approach that is based on linear smoothing of the travel time function. The underlying travel speeds are considered to be piecewise constant, as given by traffic information systems. Smoothing is used to comply with the FIFO property, as such speed data usually leads to travel time functions with jumps between two time intervals. For solving a VRP with customer and route time windows, where waiting is not allowed, they have implemented various heuristics, based on combinations of a savings heuristic, sequential insertion heuristic, and a 2-opt algorithm. Their approach is tested on real data from logistics service providers in Berlin with up to 786 orders and 84 vehicles. The results are also analyzed with respect to different numbers of time slots. More recent work on this topic has been done for example by Kritzinger et al. (2011). The authors are tackling a TDVRP with soft time windows by using a Variable Neighborhood Search procedure. Ehmke et al. (2012) are presenting an optimization framework to solve the time-dependent traveling salesman problem (TDTSP) and the TDVRP. The framework is based on several simple heuristics, as well as on a TS based metaheuristic. In addition to optimization, their work also focuses on data collection and preparation.

Optimizing the start time of a route has a major impact on the objective function, due to the tradeoff between the travel time and the tardiness from time windows. Even though, this is rarely discussed within current work which is based on heuristic or metaheuristic solution approaches. Dabia et al. (2013) claim to be the first, solving the TDVRP with time windows exactly, using a Branch-and-cut-and-price algorithm. Time-dependency is modeled in the same way as in Ichoua et al. (2003), stepwise speed functions are translated into stepwise linear travel time functions to satisfy the FIFO principle.

3 Problem description

The HHC problem we are dealing with is based on the demands of the ARC and has a daily planning horizon. It specifically aims to deal with the peculiarities of urban regions and can be described by a given set of care staff members $\mathcal{N}$ and a given set of clients $\mathcal{C}$. Each client $c \in \mathcal{C}$ needs one or more services per day, while each service is modeled as a single job $j \in \mathcal{J}$. Depending on the task that has to be carried out, a certain minimal qualification level q_j is needed. For example, if the job consists of cleaning or preparing lunch it is carried out by home helpers, whereas for medical treatments a nurse is needed. It is assumed that staff members are allowed to carry out jobs that require a qualification level that is one level below their qualification. The duration d_j of a job j is of fixed length and the job must start within a given hard time window bounded by $[a_j, b_j]$. A smooth communication and mutual trust are crucial for successful care. Therefore, the language skills of the staff members and clients, as well as the preferred or rejected (e.g., due to previous incidents or personal dislike) care staff members of the clients have to be considered. Along with the qualification, this information limits the assignment of jobs to staff members. Rather than taking care of each restriction individually, we

Table 1 Notations for the model formulation

Data	
N	Number of care staff members
S	Number of shifts (tours)
C	Number of clients
J	Number of jobs
L	Number of qualification levels
T	Number of discrete time values during the working day
P	Number of lengths in which the break can be partitioned
$\mathcal{N} = \{1, \ldots, N\}$	Set of care staff members
$\mathcal{N}_j^F \subseteq \mathcal{N}$	Set storing the feasible care staff member for each job j
$\mathcal{S} = \{1, \ldots, S\}$	Set of shifts
$\mathcal{C} = \{1, \ldots, C\}$	Set of clients
$\mathcal{J} = \{1, \ldots, J\}$	Set of jobs
$\mathcal{J}_0 = \{1, \ldots, J+1\}$	Set of jobs including the artificial depot
$\mathcal{L} = \{1, \ldots, L\}$	Set of qualification levels
$\mathcal{T} = \{1, \ldots, T\}$	Set of discrete time values
$\mathcal{P} = \{p_r : r \in \{1, \ldots, P\}\}$	Indexed set of break lengths
W_n	Contracted working time of care staff member n
$\bar{W}_n$	Maximum daily working time of care staff member n
ST	Time after which a waiting time leads to a second shift
SC	Compensation for working a second shift
OC	Compensation for overtime
D	Point of time marking the start of evening shifts
MS	Minimum working time for morning shifts
ES	Minimum working time for evening shifts
PT	Working time after which a break is mandatory
PR_k	Required break time if the care staff member of shift k needs to make a break
O	Maximum amount of overtime over all care staff members
q_j	Qualification level of job j
Q_n	Qualification level of care staff member n
d_j	Service time for a job j
a_j, b_j	Lower and upper bound of the time window of job j
A_k, B_k	Lower and upper bound of the working time window of a care staff member's shift k according to the roster
$\bar{A}_k, \bar{B}_k$	Lower and upper bound of the working time window of a care staff member's shift k according to legal restrictions
tt_{ijt}	Travel time from job i to job j departing at time t
$\vartheta : \mathcal{S} \rightarrow \mathcal{N}$	Function matching shifts with corresponding staff members

Decision variables	
x_{ijtk}	Binary, 1 if the care staff member departs from job i at time t in order to visit job j on tour k, 0 otherwise
y_{jrk}	Binary, 1 if a break of length r is made before job j on tour k, 0 otherwise

Table 1 continued

Decision variables	
Y_k	Binary, 1 if there is a break on tour k, 0 otherwise
z_n	Binary, 1 if care staff member n works a second shift, 0 otherwise
$v_k, \bar{v}_k$	Binary, 1 if shift k is used as an morning shift (v_k) or evening shift ($\bar{v}_k$), 0 otherwise
u_j, s_j	Arrival time (u_j) and starting time (s_j) of job j
S_k, E_k	Starting (S_k) and ending time (E_k) of shift k
$o_n, \bar{o}_k$	Overtime of care staff member n (o_n) and at shift k ($\bar{o}_k$)

aggregate them into single sets $\mathcal{N}_j^F$, storing the feasible care staff member for each job j.

Each staff member $n \in \mathcal{N}$ is mainly characterized by his/her qualification level Q_n and working times. In Austria, HHC service providers are legally obliged to inform their employees some weeks in advance of their upcoming working times by handing over a roster. The roster states the begin A_k and end of each shift B_k and deviations from the rosters imply overtime. The working time of each staff member starts at the first and ends at the last client of his/her tour. As the majority of the jobs are piling up in the morning, at noon, and in the evening, they often have to work two shifts a day. Working a second shift is not very popular with the staff members because in general, they are not able to make good use of the interruption. From a modeling point of view we treat each shift $k \in \mathcal{V}$ as a single tour, using a matching function $\vartheta(k)$ to identify the corresponding staff member. Due to various contracts, staff members have not only different contracted hours W_n, but also different legal claims regarding working times (earliest working time $\bar{A}_k$, latest working time $\bar{B}_k$, minimum working time for morning shifts MS and evening shifts ES, as well as maximum total working times $\bar{W}_n$), and break times PR_k. To substitute waiting time, it is possible to split the break into smaller parts. The possible break lengths are given in the indexed set $\mathcal{P}$. What mainly distinguishes our problem from previous work in HHC is that care staff members use public transport. In urban regions like Vienna, the denser public transport infrastructure together with shorter distances between the clients encourage them to use public transport. This makes sense especially when the traffic and parking situation are taken into account. Within the public transport system staff members have to switch between different lines of buses, trams, subways, and trains to reach their destinations. These are operating on timetables and as there are severe fluctuations in the driving intervals, time-dependent travel times are considered to obtain viable solutions. Especially in the suburbs, there are severe fluctuations of travel times between rush hours and off-peak times. The travel time between two jobs i and j at a certain departure time t is given by tt_{ijt}.

The following model formulation is presented in order to give a precise problem description. We tried solving small instances with 6 staff members and 30 jobs with the solver software FICO Xpress 7.0, but due to the large number of time intervals ($T = 960$) we are not able to get a solution with an acceptable gap within several days of computation time. To solve real-world sized instances, we therefore propose solution approaches based on the metaheuristic TS as introduced in Sect. 5. Table 1 gives an overview on the notation. The binary decision variable $x_{ijtk} = 1$ if the care staff member of shift k departs from job i at time t to carry out job j, 0 otherwise. The starting times of the jobs are denoted by s_j and take discrete values from set $\mathcal{T}$. Similarly, u_j represents the starting time of a break at job j and the length of the breaks is encoded in the binary variable y_{jrk}. $y_{jrk} = 1$ if the staff member of shift k makes a break of length r before serving the client of job j. S_k (resp. E_k) represents the starting (resp. ending) time of shift k. The overtime of a staff member is counted by o_n. If a staff member works a second shift the binary variable $z_n = 1$. The starting time of a shift indicates if it has to be treated as a morning or an evening shift. If the binary variable $v_k = 1$ the staff member is working a morning shift, whereas $\bar{v}_k = 1$ indicates an evening shift, which has a larger minimum working time requirement. In case $v_k = \bar{v}_k = 0$, the shift is not performed at all.

$$\min \sum_{k \in \mathcal{S}} (E_k - S_k - PR_k \cdot Y_k) \tag{1}$$

$$+ OC \sum_{n \in \mathcal{N}} o_n \tag{2}$$

$$+ SC \sum_{n \in \mathcal{N}} z_n \tag{3}$$

$$+ \sum_{i \in \mathcal{J}_0} \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{S}: q_j < Q_{\vartheta(k)}} d_j \cdot x_{ijtk} \tag{4}$$

The main objective is to minimize the sum of all shift lengths (1), resulting in a minimization of travel and waiting times as the durations of the jobs are fixed. To ensure the satisfaction of clients and staff members a dual strategy is applied. Factors with easily determinable impacts to the objective are integrated by using a weighted objective function. Overtime (2), the number of second shifts (3), and overqualification (4) are taken into account this way. All of these are not only costly for the HHC service providers, they also have a negative impact on the satisfaction of the staff members. To comply with the objective function, those factors are also measured in minutes. Overtime is measured by the time exceeding the daily working hours or, if the optimization is based on a given roster, by the deviations from the roster. It is penalized with an overtime surcharge OC. The number of second shifts gets multiplied with the time SC a staff member gets as compensation for the double journey, while overqualification counts the time a staff member with a higher qualification level performs a job that only requires a lower qualification. Other factors are implemented through aspiration levels. This is either done by adding constraints or by tightening given hard constraints. The given time windows for jobs are quite long for HHC providers in reality. But especially aged people require consistency with respect to visiting times. Therefore, dispatchers could decide to tighten some time windows based on former visiting data in order to satisfy these demands.

$$\sum_{i \in \mathcal{J}_0} \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{S}} x_{ijtk} = 1 \quad \forall j \in \mathcal{J} \tag{5}$$

$$\sum_{j \in \mathcal{J}_0} \sum_{t \in \mathcal{T}} x_{0jtk} = 1 \quad \forall k \in \mathcal{S} \tag{6}$$

$$\sum_{i \in \mathcal{J}_0} \sum_{t \in \mathcal{T}} x_{i0tk} = 1 \quad \forall k \in \mathcal{S} \tag{7}$$

$$\sum_{i \in \mathcal{J}_0} \sum_{t \in \mathcal{T}} x_{ijtk} = \sum_{h \in \mathcal{J}_0} \sum_{t \in \mathcal{T}} x_{jhtk} \quad \forall j \in \mathcal{J}, \forall k \in \mathcal{S} \tag{8}$$

$$\sum_{i \in \mathcal{J}_0} \sum_{t \in \mathcal{T}} x_{ijtk} = 0 \quad \forall j \in \mathcal{J}, k \in \mathcal{S} : \vartheta(k) \notin \mathcal{N}_j^F \tag{9}$$

(5) guarantee that each job is covered exactly once. Even if the computed schedule may not need all care staff members, all of them leave the depot exactly once (6)–(7). However, those who are not working, finish their tour immediately at the depot ($x_{00tk} = 1$). To identify the start and end of a tour an artificial depot is used. The flow conservation constraints are given by (8), while (9) allow only those staff members to execute a job, who are feasible with respect to the qualification level, language skills, and acceptance (no rejection by the client and/or staff member).

$$s_i = \sum_{t \in \mathcal{T}} (t - d_i) \sum_{j \in \mathcal{J}_0} \sum_{k \in \mathcal{S}} x_{ijtk} \quad \forall i \in \mathcal{J} \tag{10}$$

$$s_i \geq a_i \quad \forall i \in \mathcal{J} \tag{11}$$

$$s_i \leq b_i \quad \forall i \in \mathcal{J} \tag{12}$$

$$s_i + \sum_{t \in \mathcal{T}} (d_i + tt_{ijt}) \sum_{k \in \mathcal{S}} x_{ijtk} \leq u_j + b_i \Big(1 - \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{S}} x_{ijtk}\Big) \quad \forall i, j \in \mathcal{J} \tag{13}$$

$$s_i + \sum_{t \in \mathcal{T}} (d_i + tt_{ijt}) \sum_{k \in \mathcal{S}} x_{ijtk} \geq u_j - b_j \Big(1 - \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{S}} x_{ijtk}\Big) \quad \forall i, j \in \mathcal{J} \tag{14}$$

$$u_i + \sum_{r \in \{1,\ldots,P\}} p_r \sum_{k \in \mathcal{S}} y_{irk} \leq s_i \quad \forall i \in \mathcal{J} \tag{15}$$

$$s_i - u_i - \sum_{r \in \{1,\ldots,P\}} p_r \sum_{k \in \mathcal{S}} y_{irk} \leq ST - 1 \quad \forall i \in \mathcal{J} \tag{16}$$

(10) link the start time of each job with the departure time t encoded in the binary variables x_{ijtk}, and (11) and (12) force it to be within the time window of the job. Equations (13)–(15) are responsible for setting the correct starting times of the jobs and breaks. Equation (16) limit the waiting time before each job to less than ST minutes. Waiting times of ST minutes or longer would otherwise interrupt the working time and impose another shift.

$$S_k = \sum_{t \in \mathcal{T}} t \sum_{i \in \mathcal{J}_0} x_{0itk} \quad \forall k \in \mathcal{S} \tag{17}$$

$$E_k = \sum_{t \in \mathcal{T}} t \sum_{i \in \mathcal{J}_0} x_{i0tk} \quad \forall k \in \mathcal{S} \tag{18}$$

$$S_k \geq \bar{A}_k \quad \forall k \in \mathcal{S} \tag{19}$$

$$S_k \leq E_k \quad \forall k \in \mathcal{S} \tag{20}$$

$$E_k \leq \bar{B}_k \quad \forall k \in \mathcal{S} \tag{21}$$

$$\bar{W}_n \geq \sum_{k \in \mathcal{S}: \vartheta(k) = n} (E_k - S_k - PR_k \cdot Y_k) \quad \forall n \in \mathcal{N} \tag{22}$$

$$v_k = \sum_{t \in \mathcal{T}: t < D} \sum_{i \in \mathcal{J}} x_{0itk} \quad \forall k \in \mathcal{S} \tag{23}$$

$$\bar{v}_k = \sum_{t \in \mathcal{T}: t \geq D} \sum_{i \in \mathcal{J}} x_{0itk} \quad \forall k \in \mathcal{S} \tag{24}$$

$$MS \cdot v_k + ES \cdot \bar{v}_k \leq E_k - S_k - PR_k \cdot Y_k \quad \forall k \in \mathcal{S} \tag{25}$$

(17) and (18) set the starting and ending times of the shifts. The compliance with the earliest and latest working times is ensured by (19)–(21), while observance of the maximum daily working time restriction is guaranteed by (22). Equation (23)–(24) identify if the shift is a morning or an evening shift, depending on whether the shift starts before time D or not. Subsequently, constraints (25) force that the right minimal working time is taken into account, but only if the staff member actually works that shift.

$$\begin{aligned} \bar{o}_k &\geq E_k - B_k && \forall k \in \mathcal{S} \\ o_n &= \sum_{k \in \mathcal{S}: \vartheta(k) = n} \bar{o}_k && \forall n \in \mathcal{N} \end{aligned} \tag{26a}$$

$$o_n \geq \sum_{k \in \mathcal{S}: \vartheta(k) = n} (E_k - S_k - PR_k \cdot Y_k) - W_n \qquad \forall n \in \mathcal{N} \tag{26b}$$

$$\sum_{n \in \mathcal{N}} o_n \leq O \tag{27}$$

The overtime is computed either by (26a) or (26b), depending on the desired operative scenario. Equation (26a) is used if a predefined roster is given, otherwise (26b) is applied. In the first case, overtime occurs immediately if a staff member works outside of his/her roster. A staff member may exceed his/her planned end of shift B_k, while starting earlier is not permitted. Without a given roster, only the contracted working time for each staff member W_n is considered (26b). Equation (27) restricts the total amount of overtime for the day over all care staff by the parameter O.

$$z_n \geq \sum_{k \in \mathcal{S}:\vartheta(k)=n} (v_k + \bar{v}_k) - 1 \quad \forall n \in \mathcal{N} \tag{28}$$

$$S_h \geq E_k + ST \qquad \forall k, h \in \mathcal{S} : k < h \wedge \vartheta(k) = \vartheta(h) \tag{29}$$

$$\begin{aligned} &S_h \geq E_k + tt_{ijt}(x_{i0tk} + \sum_{l \in \mathcal{T}} x_{0jlh} - 1) \\ &\quad \forall i, j \in \mathcal{J}, t \in \mathcal{T}, k, h \in \mathcal{S} : k < h \wedge \vartheta(k) = \vartheta(h) \end{aligned} \tag{30}$$

In order to count the second shifts, the binary variable z_n is set to 1 if a staff member works two shifts (28). In such a case, it has to be ensured that the time between the two shifts exceeds at least ST minutes (29), otherwise it counts as waiting time from a legal point of view. Furthermore, the time between the two shifts must be sufficient to travel from the last job of the first shift to the first job of the second shift (30).

$$Y_k \geq (E_k - S_k - PT - PR_k \cdot Y_k)/PT - 1 \quad \forall k \in \mathcal{S} \tag{31}$$

$$Y_k \leq (E_k - S_k - PR_k \cdot Y_k)/PT \quad \forall k \in \mathcal{S} \tag{32}$$

$$u_i \leq PT + S_k + b_i(1 - \sum_{r \in \{1,\ldots,P\}} y_{irk}) \quad \forall i \in \mathcal{J}, k \in \mathcal{S} \tag{33}$$

$$\sum_{r \in \{1,\ldots,P\}} y_{irk} \leq 1 - \sum_{t \in \mathcal{T}} x_{0itk} \quad \forall i \in \mathcal{J}, k \in \mathcal{S} \tag{34}$$

$$\sum_{i \in \mathcal{J}} \sum_{r \in \{1,\ldots,P\}} y_{irk} \cdot p_r = PR_k \cdot Y_k \quad \forall k \in \mathcal{S} \tag{35}$$

$$\sum_{r \in \{1,\ldots,P\}} y_{jrk} \leq \sum_{i \in \mathcal{J}} \sum_{t \in \mathcal{T}} x_{ijtk} \quad \forall j \in \mathcal{J}, k \in \mathcal{S} \tag{36}$$

$$x_{ijtk} \in \{0,1\} \quad \forall i, j \in \mathcal{J}_0, t \in \mathcal{T}, k \in \mathcal{S} \tag{37}$$

$$y_{irk} \in \{0,1\} \quad \forall i \in \mathcal{J}, r \in \{1,\ldots,P\}, k \in \mathcal{S} \tag{38}$$

$$v_k, \bar{v}_k \in \{0,1\} \quad \forall k \in \mathcal{S} \tag{39}$$

$$Y_k \in \{0,1\} \quad \forall k \in \mathcal{S} \tag{40}$$

$$z_n \in \{0,1\} \quad \forall n \in \mathcal{N} \tag{41}$$

$$s_i, u_i \geq 0 \quad \forall i \in \mathcal{J} \tag{42}$$

$$S_k, E_k \geq 0 \quad \forall k \in \mathcal{S} \tag{43}$$

$$o_n, \bar{o}_k \geq 0 \quad \forall n \in \mathcal{N} \tag{44}$$

A break has to be scheduled if the working time of a staff member exceeds a given amount of time PT (31)–(32). Furthermore, the break must start no later than PT minutes after the start of the shift (33). Equation (34) limit the number of breaks before each job to at most one and prevent that a break is made before the first job. Equation (35) guarantee that the single parts of the break sum up to the required break time PR_k of the staff member, if a break is required. To ensure that a staff member only makes a break at the jobs on his/her tour, (36) link the corresponding decision variables. The model is completed by the binary and non-negativity constraints (37)–(44).

4 Travel time modeling

The calculation of (time-dependent) shortest paths is a field of research that has been well researched. Given the time-dependent travel times for each arc, one can apply modified versions of standard algorithms for the Shortest Path Problem (SPP). Such algorithms are for example based on label setting or label correcting (e.g., Dijkstra or A*). A huge compression and classification of different algorithms for the SPP is given in Klunder and Post (2006); they implemented a total of 168 versions. Chabini and Dean (1999) compare different algorithms for the time-dependent SPP, each tested with FIFO and non-FIFO data. They also describe an algorithm called DOT (decreasing order of time), which is based on dynamic programming.

The idea behind DOT is used to efficiently compute time-dependent travel times out of the timetables of public transport service providers. Additionally, time-independent walking times TT_{ij} between clients and stations are computed out of OpenStreetMap data using geographic information systems (ArcGIS 10). This source has been chosen because the data usually contains more informal paths (e.g., through residential blocks) than commercial data and thus, it is more suitable for urban regions. The dynamic programming approach works as follows:

Let $G = (V, A)$ be a complete directed graph with a set of vertices $V = \{0, 1, \ldots, m\}$ and a set of arcs $A = \{(i,j) | i,j \in V, i \neq j\}$. $A_{it} \subseteq A$ contains all outgoing arcs from node i at time $t \in T$, and $|(i, j)|$ denotes the travel (ride) time needed to traverse the arc from vertex i to j. At different times $t \in T$, the arcs between the same vertices might have different lengths. While for the routing of the care staff only the vertices of the clients are needed, V contains both, the locations of the clients as well as the locations of the stations of the public transport system. The shortest travel time tt_{ijt} from node i to node j departing at time t can be computed by Algorithm 1.

Algorithm 1 Time-dependent travel time computation

```
tt_ijt = TT_ij   ∀ i, j ∈ V, t ∈ T
for t = T-1 → 0 do
   for all i ∈ V do
      for all (i, j) ∈ A_it do
         for all k ∈ V do
            tt_ikt = min(tt_ikt, |(i, j)| + tt_jk,t+|(i,j)|)
         end for
      end for
   end for
end for
```

The idea is as simple as efficient. First, all travel times are initialized with the time-independent travel times for walking. Afterwards, as in Chabini and Dean (1999), Algorithm 1 iterates through all time intervals, starting at the latest possible. At each time t and for each vertex i all outgoing arcs (i, j) are checked if using them leads to a shorter travel time in order to reach vertex k, compared to the shortest travel time from i to k that is already known at this time. Using this approach allows to compute a travel time matrix for each minute of the day. Thus, our modeling of the time-dependent travel times is based on a discrete time approach, but using 1 min as smallest unit, lets the travel time function behave like a discontinuous piecewise linear function.

Figure 1 shows a typical travel time function for public transport, together with the corresponding arrival time function. In this example a first service (e.g., a bus) leaves the station at a departure time $dt = 5$ and needs 5 min to reach its destination. If one arrives earlier at this station, waiting time occurs at the station. Thus, the total travel time consists of the waiting time plus the ride time. For example, leaving at $dt = 6$ generates 4 min waiting time and 5 min ride time. After $dt = 10$ the service interval is extended from 5 to 10 min while at $dt = 30$ it is shortened again to 6 min. Also the ride time is reduced to 3 min. While most of the time-dependency in public transport is caused by the varying intervals of the services, there might also be some fluctuations in ride times as well. For example, at certain times of the day it may happen that buses drive different routes between two stations in order to avoid traffic. The resulting arrival time function shows that our approach complies with the FIFO condition; hence, passing is not an issue. For several modes of the public transport system passing is also impossible by nature (e.g., trains, trams).

5 Solution approach

The planning of HHC services is characterized by an initial planning, usually a few days in advance, and some short term changes during the day of operation. While the initial planning is usually not time-sensitive, disruptions may occur several

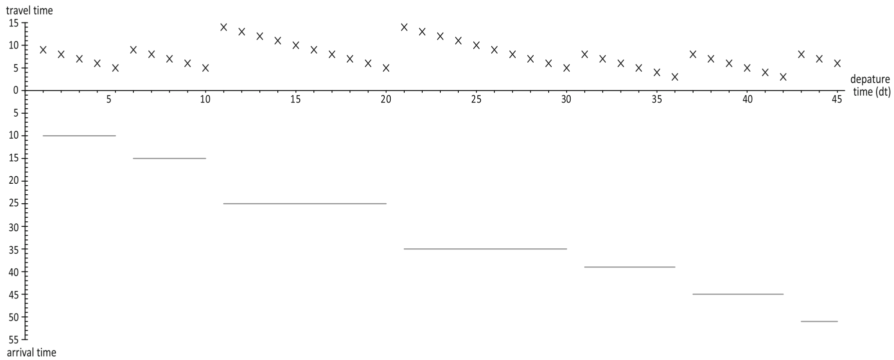

Fig. 1 Example for a typical travel and arrival time function for public transport

times a day and require rapid rescheduling. For example, if a care staff member reports sick in the morning all of his/her jobs have to be reallocated to the remaining staff. On the other hand, it is also possible that a client no longer has to be visited this day (e.g., due to hospitalization or family care), or that a new client enters the HHC system (e.g., hospital discharge). To achieve short computation times and good solution quality for real-world sized instances, we have developed three TS based solution approaches. TS was first introduced by Glover (1986) and has since been efficiently applied to various routing problems. A more recent publication about the structure of TS based metaheuristics can be found in Glover et al. (2007), together with an overview of innovative design strategies.

Algorithm 2 TS/TSAS/TSDYN: General structure

```
currentSolution s = initialSolution()
if s == feasible then
   bestSolution s* = s
   σ_jk = f(s) ∀ (j,k) ∈ s
end if
while !terminate do
   bestNeighbor s° = ∞
   bestMoves M = s.estimateMoves()
   for all j ∈ M do
      for all k ∈ S do
         for i = 0 → route[k].end() do
            build new solution s'
            move j to s'.route[k][i]
            s'.route[k].optimizeStart()
            while s'.route[k].length > PT ∧ s'.route[k].break < PR_k do
               s'.route[k].insertBreak()
               s'.route[k].optimizeStart()
            end while
            s'.evaluate()
            s'.checkTabu()
            if s' == tabu then
               s'.checkAspiration()
            end if
            if s'.objective < s°.objective then
               s°.objective = s'.objective
            end if
         end for
      end for
   end for
   s = s°
   if s == feasible ∧ s.objective < s*.objective then
      s* = s
   end if
end while
return s*
```

The main principle of TS consists of a local search combined with memory structures to guide the search and to avoid getting stuck in local optima. Starting with an initial solution, in each iteration TS switches to the best solution within a neighborhood as long as it is not flagged as 'tabu' and therefore prohibited, if it does not meet the aspiration criteria. Our algorithms are based on the ideas of the unified TS as used in time-independent routing problems (Cordeau et al. 2001; Hirsch 2011). Thus, infeasible solutions are temporarily allowed and a dynamically adapted weighted objective function is used to guide the search process. In total, we designed and implemented three different versions of time-dependent TS algorithms, which differ in the size of the neighborhood and are completely deterministic by design. Their general structure is roughly the same and outlined in Algorithm 2. The first version, simply named TS, searches the whole neighborhood in each iteration, the second uses an alternating strategy (TSAS), while the third dynamically adjusts the size of the neighborhood (TSDYN). The reason for developing different versions is the computational complexity of the evaluation of a solution and the intention to reduce the computation time. The starting time of a route not only influences the total amount of waiting time but also the travel times, due to the time-dependency. Thus, the optimal starting time is calculated immediately after composing a new route. In the next steps, breaks are inserted if necessary and the resulting solution is evaluated. Finally, the current and/or best solution, as well as all affected parameters are updated. This process is repeated until one of the termination criteria, the number of iterations or the maximum computation time, is reached. The important parts of our algorithms are outlined in the following.

5.1 Solution representation and initial solution

In our solution approaches a solution is represented by a set of routes, whereas each route represents a shift of a care staff member. Thus, if a staff member works two shifts a day each shift corresponds to a separate route. The routes itself are permutations of the assigned jobs and state the order in which they are carried out by the corresponding staff member. They start at the first client and end at the last client. The initial solution is based on an insertion heuristic. First, the jobs are ordered according to their centered time windows. Beginning with the earliest, the algorithm searches for the best assignments. The jobs of clients who have preferred care staff members are given preferential treatment. Following the best insertion principle, we try to insert them to every position of all preferred care staff members. If the insertion leads to a feasible solution, the assignment with the lowest objective value is accepted, otherwise the insertion of the job is postponed. In the next step the postponed jobs are inserted into the schedules of staff members who are not rejected by the client and who also have the appropriate skills and qualification level to carry out the job. Again only feasible solutions are accepted. At the end, all unassigned jobs are inserted according to the lowest objective value, which may lead to an infeasible initial solution.

5.2 Neighborhood

The neighborhood of a solution is composed by applying a relocate move operator. It extracts a job from its current route and re-inserts it into a different route. The insertion in the new route is based on best insertion. Our TS approach searches the whole neighborhood for the best neighbor solution. All jobs are therefore relocated and inserted at any position of all other routes. This also includes shifting jobs from the first to the second shift of a care staff member. The TSAS and the TSDYN, however, make use of a restricted neighborhood containing only a certain number of promising moves, sorted according to the length of the travel time required to serve a job. For example, moves affecting jobs at the beginning or at the end of a route are only weighted with the travel time for the arrival or the departure. For jobs in between, both the travel time for the arrival and for the departure are taken into account. Thus, moves that will remove long travel times are favored in the search process. The size of the restricted neighborhood is given by R_S, marking a percentage of the most promising moves. While the TSAS uses a restricted neighborhood of fixed size, the TSDYN adapts its size dynamically during the search process. Starting with a small restricted neighborhood, its size is increased in predefined steps R_Δ if there was no improvement of the best found solution for a certain number of iterations R_T. On the other side, its size is reset to its base value if a new best solution is found. However, to avoid being too myopic, both algorithms search the whole neighborhood after a certain number of iterations F_T. In order to escape local optima the algorithms make use of different memory structures. The most common is the tabu list, representing a short term memory. In our algorithms, τ_{jk} stores the iteration until a move containing a certain attribute (j, k) is not permitted. Here, j represents the job that has been moved and k is the shift, the job was assigned to, before executing the move. This prevents previously executed moves to be reversed for a certain number of iterations. The duration a move is considered tabu depends on the size of the instance. It is given by $\theta = \left\lceil (log_{10}(J \cdot S))^3 \right\rceil$, where J is the total number of jobs and S the number of shifts of an instance. The slope of this function ensures that the length of the tabu list is not too large to choke off the search process of small problem instances, but increases according to the size of the neighborhood to provide adequate durations for large problem instances. In addition to the tabu list a long term memory is used. The objective value of the best feasible solution containing the attribute (j, k) is stored in σ_{jk}. It is used as aspiration criteria for good neighbor solutions that are marked as tabu, in order to not reject promising solutions.

5.3 Evaluation and search guidance

A solution is evaluated according to the objective function given in Sect. 3. However, all of our algorithms allow infeasible solutions during the search process because of the sparse solution space. Therefore, an evaluation function $f(s) = c(s) + \sum \alpha_i \cdot d_i(s)$ is used to determine the quality of a solution s. $c(s)$ denotes the objective value as stated in the model formulation, which is extended by the sum of all kinds of weighted violations, denoted by $d_i(s)$ (e.g., violations of time windows, overtime, minimum and maximum working times, latest working time). To quantify these violations and to take into account the degrees of the violations, they are measured by the amount of time the limits or time windows are deviated. To force the search back to feasible solutions after exploring infeasible regions for a certain time, a dynamically adjusted weight α_i is assigned to each of these violations. At the end of each iteration the weights are increased or decreased, depending if the selected neighbor solution s' shows any of the violations. If $d_i(s') > 0$ then $\alpha_i = \alpha_i \cdot (1 + \delta)$, else $\alpha_i = \alpha_i/(1 + \delta)$, with δ as a positive input parameter. To diversify the search a penalty factor $p(s') = \lambda \cdot c(s') \cdot \sqrt{J \cdot S} \cdot \sum_{(j,k) \in s'} \rho_{jk}$ is added to the evaluation function $f(s')$ if the neighbor solution is worse than the current solution. As described in Cordeau et al. (2001) the penalty factor is composed of a scaling factor $c(s') \cdot \sqrt{J \cdot S}$ which depends on the size of the instance. The frequency based memory ρ_{jk} counts how often the attribute (j, k) was already part of a current solution in order to penalize solutions with frequently recurring attributes. Finally, a control parameter λ is used to set the intensity of the diversification.

5.4 Optimal starting times

Starting at the earliest possible starting time, which is given by the lower time window of the first job in the route, is optimal with respect to the time window violations. However, this is obviously suboptimal as it results in unnecessary waiting times at the jobs as well as for public transport and it might also generate overtime. In addition, because of the working time constraints, solutions without optimized starting times might also be infeasible. In case of time-independent travel times the optimal starting time can be efficiently determined by applying the concept of the forward time slack (45) as described in Cordeau and Laporte (2003), which is an extension of Savelsbergh (1992) in such a way that it also allows to delay the start of the route even if there are already time window violations at some jobs.

$$F_i = \min_{i \le j \le l} \left(\sum_{i < p \le j} w_p + \max(0, b_j - s_j) \right) \tag{45}$$

Given a route with a fixed order of l jobs, the slack at each job j can be computed by summing up the waiting times w_p up to job j and the positive difference between the end of the time window and the beginning of job j. The forward time slack F_i for a job i is then obtained by taking the minimum of the slacks of all following jobs j. Computing the forward time slack can be done with a complexity of $\mathcal{O}(n)$ as a whole iteration through the route is required, determining the optimal starting time can be done afterwards in $\mathcal{O}(1)$.

For time-dependent travel times the forward time slack cannot be computed that easily because the effect of a delay cannot be predicted efficiently. In contrast to the time-independent case, a delayed starting time of Δ minutes at the first job does not always result in a delay at the following jobs of $\le \Delta$ minutes but can also be $> \Delta$ if the travel time increases in the meantime. However, it is possible to compute the latest arrival time b_i^{max} that will not cause any further time window violations for each job within $\mathcal{O}(n)$.

$$b_i^{max} = \max(0, b_i - s_i) \tag{46}$$

$$b_i^{max} = \max(a_i, \min(b_i, b_{i+1}^{max} - tt^A_{i,i+1,b_{i+1}^{max}} - d_i)) \tag{47}$$

$$F_i = \max(0, b_i^{max} - (s_i - w_i)) \tag{48}$$

By iterating through the routes backwards, (46) determines the latest arrival time for the jobs at the end of the route and (47) for all other jobs. For the last job, this corresponds to the slack between the upper time window and the current starting time of the last job. For all other jobs in the route, according to (47), the time window of job i is compared with the previously computed latest arrival time of its immediate successor, denoted by $i+1$. Prerequisite is an additional set of time-dependent travel time matrices tt^A_{ijt}, storing the shortest travel time from job i to j if arrival should be at time t, which can be computed out of tt_{ijt}. The forward time slack is then given by (48). Due to its complexity of $\mathcal{O}(n \cdot T)$, determination of the optimal starting time by varying the starting time in small steps is usually done as a post optimization, as proposed for example by Fleischmann et al. (2004). However, it is possible to reduce the computational effort to $\mathcal{O}(n \cdot log\ T)$ by applying a binary search in the interval $[a_0, b_0^{max}]$ if the following conditions are met. First, the arrival time function has to be a monotonically increasing function, which is ensured by the FIFO-property, if it is satisfied. Thus, as long as postponing the start of the route does not change the end time of the route, the search is continued in the second half of the interval, otherwise in the first. Second, the evaluation function needs to be monotonic as well and directly proportional to the route lengths. This guarantees that minimizing the routes' length also minimizes the other factors in the evaluation function, if they are influenced by the start time of the route. Whereas the first condition is met, the second holds with one exception. While all other parts of the evaluation function are either not influenced by this procedure (e.g., overqualification, number of second shifts, time window violations) or directly proportional (e.g., overtime, maximum working time), only the minimum working time constraints are conflicting. However, it is debatable whether this exception is a limitation or a desired behavior from the dispatchers perspective. As the route length reduces, it might happen that the minimum working time gets violated and the previously feasible solution is now marked as infeasible. On the other hand, it would mean that a solution only satisfies the minimum working time constraint because of the insertions of waiting times and such solutions are usually rejected by the dispatchers.

Insertion of breaks If a staff member works more than a certain amount of time she/he has to make a mandatory break. In our approach breaks are inserted iteratively using the following rule. Within sequential order we try to insert the largest break type $p \in \mathcal{P}$ before job i…

1. …with largest waiting time, where $p \le w_i$.
2. …with largest waiting time, where $p \le F_i$.
3. …with largest forward time slack, where $p \le F_i$.

First we try to find a position where the waiting time covers the break at whole (1). If the waiting time is not sufficient, we are looking for the largest waiting time where an insertion is still feasible, according to the forward time slack (2). In this way we are still able to reduce the waiting time. If this is not possible either, we only try to find a feasible position (3), otherwise we add the break in the middle of the route. As it is possible to partition the break in smaller parts, we have to evaluate the route after each insertion and check if further parts are required.

6 Numerical studies

To show the applicability of the proposed solution approaches several computational experiments with real-world data from the ARC in Vienna have been carried out. For privacy reasons, it must be mentioned that the data were made available to us encoded, such that no direct reference to individuals is possible. In this paper two operative planning scenarios were analyzed. The first is based on predefined rosters and might be used for short term rescheduling. In this scenario care staff members are scheduled according to their rosters, which define their actual working times as well as possible second shifts. The second is based on flexible working times and is intended to be used for deriving a roster in mid-term. Therefore, the total working time is only limited by the contracted working time of the staff members and it is assumed that each staff member is able to work a second shift. Thus, it is up to the optimization to determine the most suitable working times for the care staff.

6.1 Real-world data sets

Two different data sets have been used for these computations. Both sets are taken from actual planning data spanning a whole week and ordered by working groups. For each day and group, the data include both, the rosters of the staff members, as well as a list of jobs that must be performed by a staff member of this group. The first data sets ($I_1 - I_7$) were collected earlier in time and cover only one geographical test area in the south of Vienna. Due to the fact that the service areas of the groups are not congruent in their entirety with the test area, not all relevant information about the whole working day of the staff members are available. It happens that staff members serve clients who are located outside the test region, or that jobs have to be scheduled which are usually carried out by staff members from another group that is not included in the data. Therefore, this data set is not used for the real-world comparison of our results with the actual planning at the ARC. To overcome this shortcoming, data for whole Vienna has been collected later on. However, this second provided data set ($I_8 - I_{20}$) contains only one qualification level. In total, three qualification levels ($Q_1 - Q_3$) are considered. The lowest level Q_1 comprises assistance in housekeeping, while the highest level Q_3 includes medical services of nurses. The service times strongly depend on the qualification level. As shown in Table 2, a job of level Q_1 ranges from 15 to 165 min, with a mean of 51.1 min, while a Q_3 job requires only 23.9 min on average. Nevertheless, the service times of the jobs are well known in advance as they are

Table 2 Data characteristics of the service times per qualification level

	Q_1	Q_2	Q_3
Min.	15	15	10
Max.	165	90	40
Mean	51.1	47.9	23.9
SD.	24.5	12.7	9.3

defined in service contracts with the clients, depending on the clients' needs. The available working time of the staff members, however, is almost independent of the qualification of the staff member. On average, a care staff member is available for about 390 min a day.

Table 3 shows the characteristics of each instance with respect to the total number of jobs, clients, and care staff members, as well as their distribution by qualification level. On the part of the staff members, also the number of second shifts is given according to their rosters. For the first data set this number is only an approximation because of the gaps in the given schedules. Usually, a staff member of level Q_1 carries out 4 to 6 jobs per day, depending on the service time of the jobs. For higher qualified staff members this ratio is usually significantly higher. The number of services requiring a higher qualification is generally lower in urban regions, due to a denser medical infrastructure. Therefore, those staff members have a much larger service area. Unfortunately, their service area is not completely covered by our test area, resulting in a surplus of staff members of level Q_2 and Q_3 in our data. However, as overqualification is permitted, these staff members are able to carry out jobs that are one level below their qualification.

In the numerical studies, an overtime surcharge of $OC = 0.5$ for each minute of overtime and a compensation of $SC = 30$ min for each second shift is presumed. The total amount of overtime over the whole care staff is limited to $O = 300$ min. In Austria, if a continuous working time of 360 min is exceeded, a break has to be scheduled. Staff members of level Q_1 need a total break time of 30 min and members of level Q_2 and Q_3 are entitled to an additional 15 min. The maximum working time of a staff member is limited to either 8 or 9 h, depending on his/her contract. The same applies to the start (6 am) and end times of a day (8 or 10 pm), which leads to a planning horizon of 960 min. Thus, 960 asymmetric travel time matrices have been computed, using the algorithm presented in Sect. 4. A total of 1,595 stations of the public transport system was taken into account during the computation of the shortest paths between all clients, resulting in a dataset of about 27 GB. As these stations are not used during the scheduling, only the travel times between the clients are needed, and because of the rather small number of clients only a few MB of travel time data is stored in memory.

6.2 Parameter tuning

The algorithms have been implemented with C++, using MS Visual Studio 2012. For the computation of the time-dependent travel times the parallel programming API OpenMP has been used to take advantage of modern multi-core processors. On

Table 3 Data characteristics of the real-world instances

	Jobs				Clients	HHC staff				
	#	Q_1	Q_2	Q_3	#	#	Q_1	Q_2	Q_3	2nd shifts
I01	106	86	20	0	50	16	10	6	0	12
I02	115	95	20	0	57	21	14	7	0	12
I03	180	152	22	6	115	34	24	7	3	9
I04	188	162	21	5	116	42	33	6	3	5
I05	193	165	22	6	127	44	34	7	3	7
I06	193	169	21	3	119	45	33	9	3	10
I07	202	177	22	3	120	46	35	9	2	14
I08	112	112	0	0	76	28	28	0	0	11
I09	114	114	0	0	78	29	29	0	0	7
I10	122	122	0	0	85	38	38	0	0	10
I11	123	123	0	0	75	29	29	0	0	8
I12	123	123	0	0	82	37	37	0	0	12
I13	126	126	0	0	75	28	28	0	0	6
I14	131	131	0	0	81	28	28	0	0	6
I05	155	155	0	0	114	34	34	0	0	0
I16	155	155	0	0	102	35	35	0	0	4
I17	157	157	0	0	103	36	36	0	0	6
I18	166	166	0	0	112	35	35	0	0	4
I19	167	167	0	0	108	34	34	0	0	5
I20	173	173	0	0	118	35	35	0	0	8

the contrary, the TS metaheuristics were implemented as single-core applications. Depending on the TS algorithms different input parameters are needed. While the TS only needs values for $\delta \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and $\lambda \in \{0.01, 0.015, 0.02, 0.025\}$, the other versions also need parameters for the size of the restricted neighborhood $R_S \in \{0.2, 0.4, 0.6\}$, and the number of iterations after which a full neighborhood search is carried out $F_T \in \{2, 4, 6, 8, 10\}$. The TSDYN additionally requires values for R_T, the time after which the restricted neighborhood is adapted, as well as the increment $R_\Delta = 0.2$. R_T depends on the size of the instance and is given by $R_T = \lceil (log_{10}(J \cdot S)) \cdot 1.5 \rceil$, where J is the total number of jobs and S the number of shifts of an instance. The length of the tabu list is determined by $\theta = \left\lceil (log_{10}(J \cdot S))^3 \right\rceil$ for all TS variants. In order to choose the best parameter settings, all possible combinations of these values have been used to solve at least two different instances, leading to a total of 600 individual solutions. Furthermore, individual parameters were identified for each operational scenario as the solution space at given rosters contains significantly less feasible solutions. The weights for temporary violations α_i were initialized with 1.0 and capped with 10^{50}. Table 4

Table 4 Parameter selection for each TS version and for each operational scenario

	Predefined roster			Flexible working time		
	TS	TSAS	TSDYN	TS	TSAS	TSDYN
δ	0.3	0.1	0.5	0.5	0.1	0.5
λ	0.020	0.010	0.010	0.025	0.015	0.020
R_S	–	0.4	0.2	–	0.4	0.2
F_T	–	2	6	–	2	10

summarizes the chosen input parameters for each TS version and for each operational scenario.

6.3 Method comparison

In this section, the algorithms are compared with each other and with the current planning at the ARC by solving the presented instances. Comparisons with exact solutions or with benchmark instances from the literature cannot be given because of the complexity of the presented HHC problem and the absence of similar VRP instances with time-dependent travel times. As stated previously, we are currently not able to solve even small instances with 6 care staff members and 30 jobs up to optimality with the solver software FICO Xpress 7.0. However, this clearly outlines interesting future research activities in order to improve the presented model formulation.

The algorithms have been executed under Windows 7 on an Intel Core i7-4930K with 64 GB memory. Even though the CPU provides six cores, the algorithms only make use of a single core, and with respect to the memory capacity it can be denoted, that the average memory requirements are around 1 GB. Due to the different neighborhood sizes of the three TS versions, a fixed runtime of 600 seconds has been chosen to achieve a fair comparison of their performances. Furthermore, real-world applicability requires short processing times in order to cope with the rescheduling in daily business. As the design and implementation of the algorithms do not contain any stochasticity, they have been executed only once. Table 5 summarizes the results for all 20 instances for each operational scenario. It shows the deviation from the objective value of the best found solution (BFS). As the static service times at the clients count for about 85 % of the objective values in our data, they are subtracted to obtain a clear picture of the real differences between the algorithms.

With a given roster the TS clearly outperforms the other versions. It is able to find the BFS 14 out of 20 times and shows only an average deviation of 0.35 % from the objective values of the BFSs. The TSDYN performs second best, but by a considerable distance. In case of flexible working times the results are contrary. In this case, the TSDYN performs best followed by the TS. In the short term, limiting the neighborhood increases the chance of missing good solutions, but in the mid- or long term it may provide additional guidance to limit the search to promising regions of the solution space. The TSAS is the worst performing algorithm in both

Table 5 Deviation from the BFS for each TS version and for each operational scenario after a computation time of 600 s [in %]

	Predefined roster			Flexible working time		
	TS	TSAS	TSDYN	TS	TSAS	TSDYN
I01	0.16	0.00	0.73	2.91	14.73	0.00
I02	2.04	3.81	0.00	5.30	0.00	2.96
I03	0.00	1.21	4.82	3.88	0.00	1.32
I04	3.40	0.13	0.00	0.41	0.95	0.00
I05	0.00	6.82	5.19	3.80	6.10	0.00
I06	0.68	0.00	5.82	7.59	0.00	1.38
I07	0.00	1.13	3.61	2.99	4.90	0.00
I08	0.00	1.19	5.01	0.30	0.71	0.00
I09	0.00	0.98	1.58	1.20	0.00	1.20
I10	0.30	4.49	0.00	2.87	3.36	0.00
I11	0.00	0.75	7.62	1.71	4.21	0.00
I12	0.00	10.14	4.71	1.16	0.00	0.18
I13	0.00	0.83	0.74	0.72	1.23	0.00
I14	0.00	2.66	0.75	0.20	3.74	0.00
I15	0.00	0.66	3.74	0.00	3.71	0.37
I16	0.00	4.90	0.41	0.00	1.21	1.07
I17	0.40	7.77	0.00	1.45	1.83	0.00
I18	0.00	2.05	0.06	0.00	1.78	0.00
I19	0.00	0.11	1.85	0.31	3.79	0.00
I20	0.00	3.80	4.99	7.32	0.91	0.00
Mean	0.35	2.67	2.58	2.21	2.66	0.42
SD	0.86	2.89	2.47	2.36	3.40	0.78
#bfs	14	2	4	3	5	13

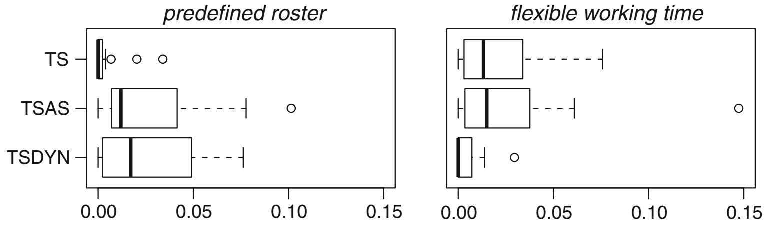

Fig. 2 Variation of deviations from the BFS for each TS version and for each operational scenario after a computation time of 600 s

scenarios, if looking at all instances. Although it finds the BFS for a few instances, the difference to the second-best solution is quite small. The TSAS needs to find a tradeoff between the computation time and the possible loss of good solutions, while dynamically adapting the size of the neighborhood allows to overcome this burden. However, the results suggest that further restricting the neighborhood within an already sparse solution space seems too myopic. With exception of the TSAS, the

Table 6 Deviation from the FFS for each TS version and for each operational scenario after a computation time of 600 s and 10 h [in %]

	Predefined roster						Flexible working time					
	TS		TSAS		TSDYN		TS		TSAS		TSDYN	
	10 min	10 h	10 min	10 h	10 min	10 h	10 min	10 h	10 min	10 h	10 min	10 h
I01	−34.50	−34.50	−27.91	−28.25	−19.13	−19.13	−53.34	−53.34	−44.10	−44.10	−49.52	−49.52
I02	−18.07	−18.07	−28.16	−28.16	−38.26	−38.44	−50.80	−51.92	−53.90	−53.90	−46.33	−46.94
I03	−33.10	−33.10	−40.08	−40.08	−38.97	−41.06	−46.63	−46.63	−46.25	−47.11	−61.05	−61.21
I04	−22.98	−22.98	−23.74	−23.74	−20.80	−20.80	−55.66	−56.37	−41.56	−41.98	−41.69	−41.84
I05	−35.96	−35.96	−31.59	−31.59	−30.62	−30.62	−48.05	−48.80	−48.36	−49.22	−38.71	−38.71
I06	−24.97	−24.97	−21.86	−21.86	−33.91	−33.91	−53.49	−54.76	−50.82	−51.44	−40.58	−40.58
I07	−48.68	−48.68	−39.90	−39.90	−44.10	−44.10	−52.60	−52.60	−41.46	−42.69	−50.13	−50.13
I08	−37.07	−37.07	−43.60	−43.60	−45.70	−45.70	−44.46	−44.78	−45.24	−45.53	−45.33	−45.56
I09	−31.74	−31.74	−37.39	−37.39	−32.60	−32.60	−56.67	−56.99	−46.77	−46.94	−39.45	−39.72
I10	−44.84	−44.84	−39.55	−39.55	−34.75	−34.75	−59.17	−59.17	−38.97	−39.21	−47.93	−47.93
I11	−52.71	−53.03	−54.44	−54.79	−37.00	−37.98	−55.31	−55.66	−51.69	−52.11	−68.34	−68.52
I12	−46.18	−46.18	−43.79	−43.79	−46.81	−46.81	−58.13	−58.31	−63.83	−63.97	−47.03	−47.09
I13	−57.48	−57.70	−30.91	−31.03	−39.36	−39.36	−59.36	−59.76	−55.76	−56.12	−44.85	−45.26
I14	−53.82	−53.82	−53.50	−53.98	−46.75	−46.83	−56.94	−57.25	−58.49	−58.88	−51.87	−52.10
I15	−61.93	−62.29	−61.29	−61.54	−51.09	−51.51	−62.80	−63.03	−62.09	−62.37	−44.12	−44.36
I16	−57.62	−57.69	−51.40	−51.81	−49.26	−49.55	−54.31	−54.72	−53.86	−54.17	−44.69	−45.11
I17	−54.47	−55.25	−54.54	−55.60	−52.38	−52.90	−43.83	−44.33	−59.86	−60.15	−48.20	−48.54
I18	−54.61	−54.92	−53.76	−54.11	−47.61	−48.16	−59.22	−59.70	−60.18	−60.75	−38.55	−38.92
I19	−48.48	−49.28	−48.66	−49.53	−46.09	−47.35	−50.39	−51.26	−46.65	−47.36	−55.39	−56.21
I20	−50.68	−51.76	−44.30	−45.77	−44.20	−45.66	−61.77	−62.80	−52.02	−52.61	−48.21	−48.80
max Δ	1.08		1.47		2.09		1.28		1.23		0.82	
mean Δ	0.20		0.28		0.39		0.46		0.44		0.26	
SD Δ	0.32		0.40		0.59		0.37		0.30		0.23	

deviations from the BFS are quite small for the TS and the TSDYN in their respective beneficial scenario. The variations of the results are also visualized in Fig. 2.

Prolonging the computation time does not improve the results considerably. In this regard, Table 6 shows the improvements of the first feasible solutions (FFS) after 10 min and after 10 h. It implies a strong relationship between the applied TS version and the operational scenario. If the algorithm performs well in the operational scenario, only marginal savings can be achieved. At the predefined roster scenario, for example, increasing the computation time of the TS to 10 h allows to increase the reported objective values by 0.20 percentage points on average and by 1.08 at most. Figure 3 exemplarily visualizes the convergence of the search for both operational scenarios for a single instance. For imaging reasons, the x-axis is scaled logarithmically and cut at 10^6 iterations, as no improvements have been achieved afterwards. However, the total number of iterations each algorithm performed within the 10 h limit is given in square brackets in the chart legend.

6.4 Real-world comparison

Table 7 shows the deviations of the objective values of our algorithms from the actual planned routes at the ARC. For this purpose, the schedules we received from the ARC are evaluated according to our objective function given in Sect. 3. The scheduled end of each job is taken as departure time for the time-dependent travel times. Any time window or working time violations that occur during the evaluation of the ARC schedule are neglected.

Although it is evident which algorithm is suitable for which operational scenario, significant savings can be achieved with all of them, if compared with the actual

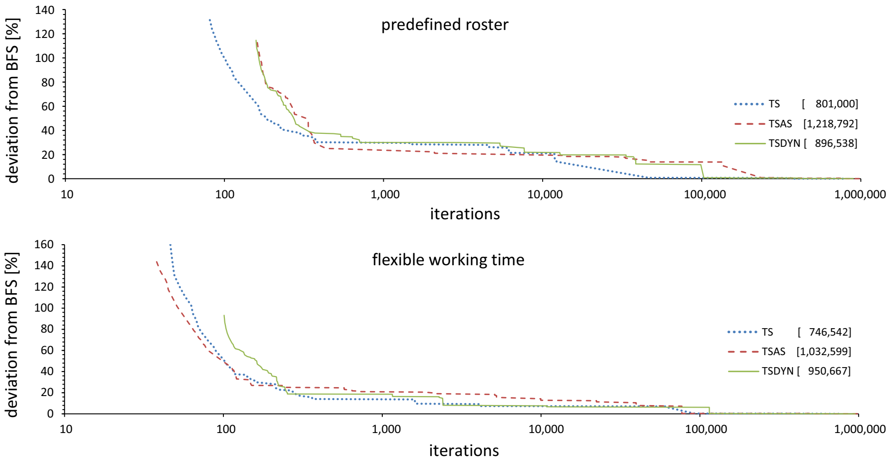

Fig. 3 Search convergence for each operational scenario of instance I13: deviation of the objective value from the BFS at all iterations achieved with a computation time of 10 h [in %]

Table 7 Deviation from actual planning at ARC for each TS version and for each operational scenario after a computation time of 600 s [in %]

	Predefined roster			Flexible working time		
	TS	TSAS	TSDYN	TS	TSAS	TSDYN
I08	−37.78	−37.04	−34.67	−50.94	−50.74	−51.09
I09	−29.82	−29.13	−28.71	−46.19	−46.82	−46.19
I10	−40.68	−38.20	−40.85	−56.15	−55.94	−57.37
I11	−51.46	−51.10	−47.76	−59.22	−58.22	−59.91
I12	−40.79	−34.78	−38.00	−52.03	−52.58	−52.49
I13	−49.25	−48.83	−48.88	−54.16	−53.93	−54.49
I14	−48.85	−47.49	−48.47	−57.79	−56.30	−57.87
I15	−61.06	−60.81	−59.61	−65.31	−64.02	−65.18
I16	−37.66	−34.61	−37.40	−40.59	−39.86	−39.95
I17	−45.72	−41.74	−45.94	−51.85	−51.67	−52.54
I18	−34.85	−33.51	−34.81	−38.70	−37.60	−38.70
I19	−35.31	−35.25	−34.12	−43.99	−42.04	−44.16
I20	−26.63	−23.84	−22.97	−39.96	−43.55	−44.06
mean	−41.53	−39.72	−40.17	−50.53	−50.25	−51.08
SD	9.50	10.02	9.79	8.20	7.83	8.04

planning of the ARC. This is explained by the short travel times in urban regions combined with the fact that a compensation in the amount of 30 min working time is accounted for a second shift. In almost each instance, it was possible to significantly reduce the number of second shifts or even to avoid them entirely. As those shifts are quite uncomfortable for most staff members, this result has a major impact on their satisfaction. If using the TS, savings between 26.63 % and 61.06 % are achievable, with a mean of 41.53 %. If not limited by a given roster, it is possible to further increase the savings by an additional 10 percentage points, on average.

7 Conclusion and outlook

We present a mathematical model for the daily scheduling of HHC services, where care staff members are using public transport with time-dependent travel times. The time-dependent travel times are computed by a dynamic programming approach, which is able to compute travel time matrices for each minute of the day out of timetable data. Although this leads to the most accurate travel times, it also revealed a drawback of our modeling approach of using minute-based travel times. Due to the large number of time intervals, it was not possible to get an exact solution with an acceptable gap within reasonable time for a small test instance with 6 staff members and 30 jobs, using the solver software FICO Xpress 7.0. Nevertheless, minute-based travel times are necessary for scheduling with public transport, especially for services in sub-urban regions or during off-peak times.

To solve real-world sized instances within reasonable computation time we developed three TS based metaheuristics. The TS always searches the whole neighborhood, the TSAS uses a restricted neighborhood of fixed size, and the TSDYN adapts the neighborhood dynamically. In case of time-independent travel times, optimizing the starting times of the routes is usually straightforward, but with time-dependent travel times, this task requires a lot of computational effort. To tackle this challenge, we use a binary search to identify the optimal time.

The approach has been tested with real-world data from the ARC in Vienna. Two data sets with a total number of 20 instances of different sizes are used to compare the algorithms with each other, as well as the actual planning at the ARC. The largest instance consists of 202 jobs and 46 staff members. Two operational scenarios are analyzed. The first is based on a predefined roster and represents daily business, where the dispatcher is legally bounded by the rosters that have been given to the care staff members a few weeks before. The second scenario assumes flexible working times for the staff members and is intended to show the full potential savings, if not limited by a given roster. It might be used to derive an optimized roster in future. However, additional constraints dealing with rest periods and overtime during a certain period would be needed in such a model. Furthermore, one has to cope with the uncertainties in the data, which arise due to the temporal distance between the drawing up of the rosters and the daily scheduling of the care staff. These must also be kept in mind at the comparison of the results of the second scenario with the planning at the ARC.

In the first scenario, the TS algorithm outperforms the other versions by almost two percentage points. In the second scenario the results are contrary, with the TSDYN outperforming the others. This behavior is explained by the sparse solution space in the first scenario. A given roster already limits the solution space in a way that an additional reduction of the neighborhood leads to unfavorable solutions. With flexible working times, however, the restricted neighborhood acts as an additional guidance into promising parts of the solution space. The dynamic adjustment of the size of the neighborhood allows to overcome the tradeoff between the computation time and the possible loss of good solutions.

If compared with the planning of the ARC, average savings of 41.53 % with a given roster, and 51.08 % without a roster are achievable through optimization. As for most real-world comparisons one has to bear in mind that the schedules with which we compared our results may not be based only on the rosters of the care staff members but also on undocumented knowledge we are not aware of. For example, it is conceivably that staff members are available throughout the day according to their rosters, but are scheduled only on the morning because of some private appointments. The same might apply in the other direction when a staff member has no duty but has to stand in for somebody who calls in sick.

In summary, it can be stated that there is a significant savings potential compared to the current routing and using time-dependent travel times result in more reliable schedules.

Day-to-day business in HHC is usually subject to stochastic and dynamic events that require repeated adjustments throughout a day. For example, the actual service time at a client depends on his/her health condition and thus, the scheduled service time might be exceeded easily. In addition, it might happen that the client has been hospitalized on short notice and the scheduled visit is canceled. Even if good schedules can be computed within seconds, the presented solution approach is currently only capable of computing schedules from scratch. Rescheduling requires an adjustment of the input data and may lead to entirely different schedules. However, it might be preferable to adapt to the new situation with as few changes as possible. Therefore, future research focuses on a new solution approach that is operational in dynamic environments, and thus able to overcome this limitations. In the mid-term, it is also planned to integrate additional independent transport modes like cycling, car driving, or pickup services (see, e.g., Fikar and Hirsch 2014). Allowing combinations of these moves the modal choice into optimization. This facilitates the consideration of user-dependent requirements (e.g., maximum cycling distance).

Acknowledgments Financial support by Grant No 835770 of the program iv2splus provided by the Austrian Research Promotion Agency and the Federal Ministry for Transport, Innovation and Technology, is gratefully acknowledged. Furthermore, are grateful to the ARC for providing data and suitable information; especially to Monika Wild and Harald Pfertner. We further want to thank Melanie Herzog from the Technische Universität München for her support during the development of the dynamic programming approach.

References

Akjiratikarl C, Yenradee P, Drake PR (2007) Pso-based algorithm for home care worker scheduling in the UK. Comput Ind Eng 53(4):559–583. doi:10.1016/j.cie.2007.06.002

Begur SV, Miller DM, Weaver JR (1997) An integrated spatial dss for scheduling and routing home-health-care nurses. Interfaces 27(4):35–48. doi:10.1287/inte.27.4.35

Bertels S, Fahle T (2006) A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem. Comput Oper Res 33(10):2866–2890. doi:10.1016/j.cor.2005.01.015

Bräysy O, Dullaert W, Nakari P (2009) The potential of optimization in communal routing problems: case studies from Finland. J Transp Geogr 17(6):484–490. doi:10.1016/j.jtrangeo.2008.10.003

Bredström D, Rönnqvist M (2007) A branch and price algorithm for the combined vehicle routing and scheduling problem with synchronization constraints. Discussion Paper No. 2007/7, NHH Department of Finance and Management Science, Norwegian School of Economics, Norway, doi:10.2139/ssrn.971726

Bredström D, Rönnqvist M (2008) Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. Eur J Oper Res 191(1):19–31. doi:10.1016/j.ejor.2007.07.033

Cappanera P, Scutellà MG (2014) Joint assignment, scheduling and routing models to home care optimization: a pattern based approach. Transp Sci 1–23. doi:10.1287/trsc.2014.0548

Chabini I, Dean B (1999) Shortest path problems in discrete-time dynamic networks: Complexity, algorithms, and implementations. Tech. rep, Massachusetts Institute of Technology, Cambridge, MA

Cheng E, Rich JL (1998) A home health care routing and scheduling problem. Tech. rep. tr98-04, Department of CAAM, Rice University, Houston Texas, USA

Cordeau JF, Laporte G (2003) A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transp Res B 37(6):579–594. doi:10.1016/S0191-2615(02)00045-0

Cordeau JF, Laporte G, Mercier A (2001) A unified tabu search heuristic for vehicle routing problems with time windows. J Oper Res Soc 52(8):928–936

Dabia S, Ropke S, Van Woensel T, De Kok T (2013) Branch and price for the time-dependent vehicle routing problem with time windows. Transp Sci 47(3):380–396. doi:10.1287/trsc.1120.0445

Dohn A, Rasmussen MS, Justesen T, Larsen J (2008) The home care crew scheduling problem. In: Sheibani K (ed) Proceedings of the 1st international conference on applied operational (ICAOR'08), Yerevan, Armenia, Lecture Notes in Management Science, vol 1, pp 1–8

Ehmke JF, Steinert A, Mattfeld DC (2012) Advanced routing for city logistics service providers based on time-dependent travel times. J Comput Sci 3(4):193–205. doi:10.1016/j.jocs.2012.01.006

Elbenani B, Ferland J, Gascon V (2008) Mathematical programming approach for routing home care nurses. In: Proceedings of industrial engineering and engineering management 2008 (IEEM08), Singapore, pp 107–111, doi:10.1109/IEEM.2008.4737841

Eveborn P, Flisberg P, Rönnqvist M (2006) Laps care-an operational system for staff planning of home care. Eur J Oper Res 171(3):962–976. doi:10.1016/j.ejor.2005.01.011

Eveborn P, Rönnqvist M, Einarsdóttir H, Eklund M, Lidén K, Almroth M (2009) Operations research improves quality and efficiency in home care. Interfaces 39(1):18–34. doi:10.1287/inte.1080.0411

Fikar C, Hirsch P (2014) A matheuristic for routing real-world home service transport systems facilitating walking. J Clean Prod. doi:10.1016/j.jclepro.2014.07.013

Fleischmann B, Gietz M, Gnutzmann S (2004) Time-varying travel times in vehicle routing. Transp Sci 38(2):160–173. doi:10.1287/trsc.1030.0062

Glover F (1986) Future paths for integer programming and links to artificial intelligence. Comput Oper Res 13(5):533–549. doi:10.1016/0305-0548(86)90048-1

Glover F, Laguna M, Marti R (2007) Principles of tabu search. In: Gonzalez TF (ed) Handbook of approximation algorithms and metaheuristics. Chapman & Hall/CRC, Boca Raton, pp 23-1–23-11. doi:10.1201/9781420010749

Hiermann G, Prandtstetter M, Rendl A, Puchinger J, Raidl GR (2013) Metaheuristics for solving a multimodal home-healthcare scheduling problem. Cent Eur J Oper Res 23:1–25. doi:10.1007/s10100-013-0305-8

Hill AV, Benton WC (1992) Modelling intra-city time-dependent travel speeds for vehicle scheduling problems. J Oper Res Soc 43(4):343–351. doi:10.2307/2583157

Hirsch P (2011) Minimizing empty truck loads in round timber transport with tabu search strategies. Int J Inf Syst Supply Chain Manag 4(2):15–41. doi:10.4018/jisscm.2011040102

Ichoua S, Gendreau M, Potvin JY (2003) Vehicle dispatching with time-dependent travel times. Eur J Oper Res 144(2):379–396. doi:10.1016/S0377-2217(02)00147-9

Klunder G, Post H (2006) The shortest path problem on large-scale real-road networks. Networks 48(4):182–194. doi:10.1002/net.20131

Kritzinger S, Tricoire F, Dörner KF, Hartl RF (2011) Variable neighborhood search for the time-dependent vehicle routing problem with soft time windows. In: Coello CAC (ed) Proceedings of the 5th international conference on learning and intelligent optimization (LION'05), Rome, Italy, Lecture Notes in Computer Science, vol 6683, pp 61–75

Liu R, Xie X, Garaix T (2014) Hybridization of tabu search with feasible and infeasible local searches for periodic home health care logistics. Omega 47:17–32. doi:10.1016/j.omega.2014.03.003

Malandraki C, Daskin MS (1992) Time dependent vehicle routing problems: formulations, properties and heuristic algorithms. Transp Sci 26(3):185–200. doi:10.1287/trsc.26.3.185

Mankowska DS, Meisel F, Bierwirth C (2013) The home health care routing and scheduling problem with interdependent services. Health Care Manag Sci 17(1):15–30. doi:10.1007/s10729-013-9243-1

Matta A, Chahed S, Sahin E, Dallery Y (2014) Modelling home care organisations from an operations management perspective. Flex Serv Manuf J 26(3):295–319. doi:10.1007/s10696-012-9157-0

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(3):598–610. doi:10.1016/j.ejor.2011.10.048

Rest KD, Trautsamwieser A, Hirsch P (2012) Trends and risks in home health care. J Humanit Logist Supply Chain Manag 2(1):34–53. doi:10.1108/20426741211225993

Savelsbergh MWP (1992) The vehicle routing problem with time windows: minimizing route duration. ORSA J Comput 4(2):146–154. doi:10.1287/ijoc.4.2.146

Trautsamwieser A, Hirsch P (2014) A branch-price-and-cut approach for solving the medium-term home health care planning problem. Networks 64(3):143–159. doi:10.1002/net.21566

Trautsamwieser A, Gronalt M, Hirsch P (2011) Securing home health care in times of natural disasters. OR Spectr 33(3):787–813. doi:10.1007/s00291-011-0253-4

Klaus-Dieter Rest is a research assistant at the Institute of Production and Logistics, University of Natural Resources and Life Sciences, Vienna, Austria. He holds a diploma in Business and Economics, specialist area Management Science and is currently working on his doctoral thesis in the field of optimization of urban home health care. His research interests are within the field of health care logistics, urban logistics but also in disaster management.

Patrick Hirsch is an assistant professor, project manager and vice head of the Institute of Production and Logistics at the University of Natural Resources and Life Sciences, Vienna. He acts as a reviewer for several international scientific journals and is associate editor of the Journal of Applied Operational Research. His research interests are within transportation logistics, health care logistics and disaster management. He presented his work at several international conferences and published some book chapters as well as scientific journal articles. For his doctoral thesis in the field of timber transport he received two awards.