A Branch-Price-and-Cut Approach for Solving the Medium-Term Home Health Care Planning Problem

Andrea Trautsamwieser and Patrick Hirsch
Institute of Production and Logistics, Department of Economics and Social Sciences, University of Natural Resources and Life Sciences, Vienna, Feistmantelstrasse 4, 1180 Vienna, Austria

The planning of home health care services is still done manually in many industrial countries. However, efficient decision support is necessary to improve the working plans and relieve the nurses from this time consuming task. The problem can be summarized as follows: clients need to be visited one or several times during the week by appropriately skilled nurses; their treatments have predefined time windows. Additionally, working time requirements for the nurses such as breaks, maximum working time per day, and daily as well as weekly rest times have to be considered. We propose a Branch-Price-and-Cut solution approach to solve this problem exactly, using the solutions of a variable neighborhood search solution approach as upper bounds. The algorithm is capable of solving to optimality real-life based test instances with up to nine nurses, 45 clients, and 203 visits during the week. **NETWORKS, Vol. 64(3), 143–159 2014**

Keywords: Branch-Price-and-Cut; home health care; vehicle routing; variable neighborhood search; weekly planning horizon

1. INTRODUCTION

In many industrial countries, the planning of home health care (HHC) services is done manually. However, due to the increase in life expectancy, changes in the social living situation, and changes in policies, the demand for HHC services rises. Therefore, efficient planning is becoming more difficult and particularly time consuming.

HHC services are typically planned 1 week ahead to inform the nurses about their work schedules for the next week. Therefore, the medium-term HHC planning problem can be stated as follows. Given a planning horizon of 1 week, find a work schedule for each nurse such that all clients are visited during this period as often as required. These visits need to be done by appropriately skilled nurses in the given time windows. Additionally, maximum working times per day, breaks during the day as well as sufficient rest times between working days have to be respected. The objective is to minimize the total working time of the nurses.

In practice, this problem is usually solved manually in a stepwise procedure. The visits at the clients are first scheduled to certain visiting days. Then, for each day, the visits are assigned to the nurses, and finally, the nurses plan the sequence of their visits. However, this method might lead to infeasible solutions that have to be modified (several times) to obtain feasible solutions. Especially, the working time restrictions such as breaks during the day, maximum working time per day, and sufficient rest times during the week make the manual planning quite complex. The introduced problem has a lot of interesting properties and to the best of our knowledge it has not been addressed in research so far.

The main contribution of this article is to present a novel mathematical model formulation for the medium-term HHC planning problem and to introduce a Branch-Price-and-Cut (BPC) solution approach for it. Additionally, we develop a metaheuristic solution method based on variable neighborhood search (VNS) to provide the BPC algorithm with good upper bounds in short computation times. The presented solution approaches and their implementations are explained in detail and tested in extensive numerical studies. The VNS method is capable to provide good upper bounds after 10^6 iterations within less than 27s of computation time. The BPC approach is able to solve to optimality instances with up to nine nurses and 45 clients leading to 203 visits during the week within a time limit of 1 h.

This article is organized as follows: in Section 2, an overview of the current literature on this problem is given. The problem formulation and the mathematical model are described in Section 3. In Section 4, the BPC solution approach is presented. Computational results are presented in Sections 5 and 6 conclude the article.

2. LITERATURE REVIEW

Some literature already deals with the daily planning problem of HHC services. This problem is a relaxation of the

Received December 2012; accepted August 2014
Correspondence to: A. Trautsamwieser; e-mail: andrea.trautsamwieser@gmail.com
DOI 10.1002/net.21566
Published online 15 September 2014 in Wiley Online Library (wileyonlinelibrary.com).

medium-term HHC planning problem as services are only planned for one particular day and rest times between working days as well as weekly rest times are not considered. However, this problem is still interesting because it is sometimes necessary to reschedule HHC services, in case of nurses getting ill or additional clients need to be serviced in the course of the week.

The following papers deal with a daily planning horizon. They differ in several assumptions as the organization of HHC services depends a lot on the legislation in the application area. To the best of our knowledge Begur et al. [2] and Cheng and Rich [11] were among the first who addressed this problem. Begur et al. [2] present a decision support system (DSS) for a case study and solve small instances with a heuristic solution approach including up to seven nurses and 40 clients. Two mathematical model formulations that consider breaks and a two phase heuristic solution approach are proposed by Cheng and Rich [11]. Real-life instances with up to 294 nurses and 900 clients are solved by their heuristic. Bertels and Fahle [3] present a hybrid solution approach consisting of simulated annealing, tabu search, and constraint programming. Instances with up to 50 nurses, 200 clients, and 600 visits are solved. In contrary, Eveborn et al. [15] propose a DSS called Laps Care. It is based on a set partitioning formulation solved by a repeated matching algorithm. Computational results with up to 21 nurses and 123 clients are presented. Real-life experiences with the system are reported in Eveborn et al. [16]. Bräysy et al. [5] use a commercial vehicle routing problem (VRP) solver to show improvement potentials for different scenarios in a case study, including 38 nurses and 765 visits a week. A particle swarm optimization-based algorithm is proposed by Akjiratikarl et al. [1]. They solve five real-life instances including 12 nurses, 50 clients, and more than 100 visits. Dohn et al. [13] present a Branch-and-Price (BP) solution approach to solve real-life instances with up to 15 nurses and 150 visits. Bredström and Rönnqvist [6] also develop a BP algorithm to solve instances with up to 16 nurses and 80 visits. They integrate temporal dependencies between the visits and synchronization constraints in their approach. They also propose a heuristic solution approach in Bredström and Rönnqvist [7]. Rasmussen et al. [26] present a mathematical model formulation including synchronization constraints and introduce a BP solution approach. They solve instances including 16 nurses and 80 visits. Trautsamwieser et al. [33] present a mathematical model formulation and a VNS-based solution approach for real-life instances with up to 75 nurses, 411 clients, and 512 visits. Their approach also includes breaks that are set only if needed to the cost-optimal positions in tours. An overview of consequences of natural disasters on the planning of HHC services and an outlook to future trends are given in Rest et al. [29]. The above stated approaches consider the use of one transport mode during the tour of a nurse. Different modes of transport are discussed in Rendl et al. [27] as well as in Rest and Hirsch [28]. Rendl et al. [27] present a hybrid solution approach combining constraint programming and a metaheuristic to solve instances with up to 509 nurses and 717 visits. Additionally, Rest and Hirsch [28] consider time-dependent traveling times for public transport. They propose Tabu Search-based solution approaches and solve instances with up to 89 nurses, 236 clients, and 388 visits.

Recently, some research groups also deal with the medium-term planning of HHC services. Gamst and Jensen [18] propose a mathematical model formulation and a BP solution approach. Their work focuses on regularity issues dealing with regular visiting times at the clients by the same group of nurses. Visiting time windows and working time windows of the nurses are not considered as hard constraints but are handled within the weighted objective function. Instances with up to two nurses and 50 visits during 1 week with a discretization time unit of 10 min are solved to optimality. Nickel et al. [25] propose a two stage heuristic solution approach. They use a hybrid constraint programming heuristic to generate initial solutions, which are further improved by an adaptive large neighborhood solution approach. They use a weighted objective function consisting of the number of unscheduled visits, the total traveling time, overtime, and regular visit combinations. Real-life instances with up to 12 nurses, 95 clients, and 361 visits are solved for a weekly planning horizon. Cattafi et al. [8] present a solution approach based on constraint programming in which they assign the nurses to the clients and solve a traveling salesman problem for each nurse and each day. Time windows do not exist and the nurses work also in an intramural health care facility (e.g., hospital) during their shift. The total workload of the nurses and the number of different nurses visiting a client are minimized. Solutions for real-life instances including 15 nurses, 458 clients, and 3323 visits during a month are compared on a weekly basis to the manually generated solutions in practice.

Beside the different methodology, a main distinctive feature of our research on the medium-term planning of HHC services is the consideration of working time restrictions. None of the approaches we are aware of is dealing with breaks or (weekly) rest times.

3. PROBLEM DESCRIPTION

HHC planning consists of several decisions. In practice, the main decision is the assignment of each visit of a client during the week to a nurse. Thereby, it has to be considered that some nurses might be rejected for several reasons (e.g., smokers) by the client or the nurse rejects the client. This information is available before the planning procedure. Additionally, the nurse needs to have at least the same or a higher qualification level as required by the treatment to be allowed to visit the client. Furthermore, the sequence of the visits has to be computed such that the clients are visited during their time windows and the working time requirements of the nurses are fulfilled. Hereby, we have to take into account that a nurse is not allowed to work longer than a given threshold L per day. Moreover, a break of length p needs to be scheduled if the working time per day is longer than B minutes. The nurses are only allowed to make one

TABLE 1. Notations for the mathematical model formulation.

Notation	Definition
J	Number of clients
V	Number of nurses
T	Number of days in the planning horizon
$\mathcal{J} = \{1, \ldots, J\}$	Set of clients
$\mathcal{J}_0 = \{0, \ldots, J+1\}$	Set of clients including the depots 0 and $J+1$
$\mathcal{V} = \{1, \ldots, V\}$	Set of nurses (tours per day)
$\mathcal{T} = \{1, \ldots, T\}$	Set of days
P	Break node
p	Break length
B	Maximum working time per day without a mandatory break
L	Maximum working time per day including a break
W	Maximum working time per week
R_{d}	Minimum daily rest time between two consecutive working days
$R_{\mathrm{w}} + 1440$	Minimum weekly rest time
Q_{jk}	Binary constant, equals 1 if client j is allowed to be serviced by nurse k, 0 otherwise
F_{jt}	Binary constant, equals 1 if client j requires service on day t, 0 otherwise
t_{ij}	Integer constant, travelling time between the single locations (clients/depots)
d_j	Integer constant, service duration at client j
a_j, b_j	Lower and upper bound of the time window of client j
a_0^{kt}, b_0^{kt}	Lower and upper bound of nurse k's working time window on day t
x_{ijkt}, x_{iPkt}	Binary decision variable, equals 1 if client j (or break P) is visited after client i by nurse k on day t, 0 otherwise
s_{jkt}, s_{Pkt}	Integer decision variable, starting time at client j or break P on tour k on day t
y_{kt}	Binary decision variable, equals 1 if a break is taken on tour k on day t, 0 otherwise

break per tour. This break can be taken at any time during the day as long as the maximum consecutive working time does not exceed B. It is further assumed that $2B > L$. The break is typically scheduled at a client, before or after the service. Further, the total working time per week is not allowed to exceed a given limit W and sufficient rest times are needed during the week. A nurse needs to rest at least R_{d} minutes between two consecutive working days. Moreover, nurses are allowed to be on duty on all 7 days of the week. To guarantee a sufficient recreational period, a rest time of at least $R_{\mathrm{w}} + 1440$ is required during the week. A plan of HHC services is efficient, if the total working time of the nurses is minimized. As the service times are constant, a minimization of the working times equals a minimization of the traveling times and waiting times. The nurses as well as the HHC providers have an interest in keeping these times short to spend most of the working time with the clients. In the following, we introduce the different variables for the mathematical model.

Each client $j \in \mathcal{J} = \{1, \ldots, J\}$ is visited once or several times a week on prefixed visiting days for a certain service. This might be an insulin shot required on each day of the week or the provision of meals. Hence, these visits are scheduled at the same time during the day. For instance, a client needs three visits a week (on Monday, Wednesday, and Friday) and all visits must be provided between 11.00 am and 1.00 pm. Performing this service on Monday is one particular visit, and further called visit. Hence, a client who requires to be treated daily generates seven visits. Each client j can be characterized by a fixed service duration d_j and a time window $[a_j, b_j]$ in which the service has to start. A nurse $k \in \mathcal{V} = \{1, \ldots, V\}$ is allowed to visit client j, if the nurse has a sufficient qualification level for meeting client j's needs and she/he is not rejected by the client or vice versa. We summarize these different properties in matrix Q. Q_{jk} equals 1 if nurse k is allowed to treat client j and 0 otherwise. Considering a planning horizon of T days, it is further possible that a nurse is available on day $t \in \mathcal{T} = \{1, \ldots, T\}$ or not. In case of unavailability, the corresponding working time window $[a_0^{kt}, b_0^{kt}]$ equals $[0, 0]$ as each day starts at time 0. Additionally, due to different work contracts, the nurse might start her tour at the HHC office, further called the depot, or at home. If she starts at home, we further have to distinguish whether the traveling times to the first client and back from the last client are paid or unpaid. For simplicity, the model is formulated with one starting depot 0 and one finishing depot $J+1$ as well as nurse-dependent traveling times from and to the depots.

For the mathematical model, we define the following decision variables: Let x_{ijkt} be a binary decision variable, that equals 1 if client j is visited directly after client i by nurse k on day t, and 0 otherwise. The starting time at client j by nurse k on day t is given by s_{jkt}. s_{0kt} and $s_{J+1,kt}$ are the corresponding starting times and finishing times of the nurses. If the working time of a nurse exceeds a working time of B minutes, the nurse is required to take a break. y_{kt} is the corresponding binary decision variable and equals 1 if nurse k must take a break on day t and 0 otherwise. s_{Pkt} equals the starting time of the break. The quality of a feasible plan for HHC is measured by the total working time of the nurses. Breaks are excluded from the objective function since they do not count as working time by law and are unpaid. The notations used in the following model are summarized in Table 1.

The mathematical formulation is given by:

$$\min \quad \sum_{k\in\mathcal{V}}\sum_{t\in\mathcal{T}}(s_{J+1,kt} - s_{0kt} - p\cdot y_{kt}) \tag{1}$$

$$\text{s.t.} \quad \mathrm{x}_{\mathrm{ijkt}} \le \min(Q_{\mathrm{ik}}, Q_{\mathrm{jk}}, \mathrm{b}_0^{\mathrm{kt}} - \mathrm{a}_0^{\mathrm{kt}}) \quad \forall i,j\in\mathcal{J}_0, \forall \mathrm{k}\in\mathcal{V}, \forall \mathrm{t}\in\mathcal{T} \tag{2}$$

$$\sum_{i\in\mathcal{J}_0}\sum_{k\in\mathcal{V}} x_{ijkt} = F_{jt} \quad \forall j\in\mathcal{J}, \forall t\in\mathcal{T} \tag{3}$$

$$\sum_{j\in\mathcal{J}} x_{0jkt} \le 1 \quad \forall k\in\mathcal{V}, \forall t\in\mathcal{T} \tag{4}$$

$$\sum_{i\in\mathcal{J}_0} x_{ijkt} = \sum_{h\in\mathcal{J}_0} x_{jhkt} \quad \forall j\in\mathcal{J}, \forall k\in\mathcal{V}, \forall t\in\mathcal{T} \tag{5}$$

$$\sum_{i\in\mathcal{J}} x_{iPkt} = y_{kt} \quad \forall k\in\mathcal{V}, \forall t\in\mathcal{T} \tag{6}$$

$$x_{Pjkt} = x_{jPkt} \quad \forall j\in\mathcal{J}, \forall k\in\mathcal{V}, \forall t\in\mathcal{T} \tag{7}$$

$$x_{Pjkt} \le \sum_{i\in\mathcal{J}} x_{ijkt} \quad \forall j\in\mathcal{J}, \forall k\in\mathcal{V}, \forall t\in\mathcal{T} \tag{8}$$

$$(s_{J+1,kt} - s_{0kt}) \le B + (L-B)y_{kt} \quad \forall k\in\mathcal{V}, \forall t\in\mathcal{T} \tag{9}$$

$$(s_{J+1,kt} - s_{0kt}) \ge (B+p)y_{kt} \quad \forall k\in\mathcal{V}, \forall t\in\mathcal{T} \tag{10}$$

$$\sum_{t\in\mathcal{T}}(s_{J+1,kt} - s_{0kt} - p\cdot y_{kt}) \le W \quad \forall k\in\mathcal{V} \tag{11}$$

$$s_{ikt} + (t_{ij}+d_i)x_{ijkt} \le s_{jkt} + b_i(1-x_{ijkt}) \quad \forall i,j\in\mathcal{J}_0, \forall k\in\mathcal{V}, \forall t\in\mathcal{T} \tag{12}$$

$$s_{Pkt} + p\cdot x_{Pjkt} \le s_{jkt} + 1440(1-x_{Pjkt}) \quad \forall j\in\mathcal{J}, \forall k\in\mathcal{V}, \forall t\in\mathcal{T} \tag{13}$$

$$s_{ikt} + (t_{ij}+d_i)(x_{ijkt} + x_{jPkt} - 1) \le s_{Pkt} + b_i(2 - x_{ijkt} - x_{jPkt}) \quad \forall i,j\in\mathcal{J}, \forall k\in\mathcal{V}, \forall t\in\mathcal{T} \tag{14}$$

$$a_j\sum_{i\in\mathcal{J}_0} x_{ijkt} \le s_{jkt} \le b_j\sum_{i\in\mathcal{J}_0} x_{ijkt} \quad \forall j\in\mathcal{J}, \forall k\in\mathcal{V}, \forall t\in\mathcal{T} \tag{15}$$

$$s_{Pkt} \le s_{0kt} + B \quad \forall k\in\mathcal{V}, \forall t\in\mathcal{T} \tag{16}$$

$$s_{Pkt} \ge s_{J+1,kt} - B \quad \forall k\in\mathcal{V}, \forall t\in\mathcal{T} \tag{17}$$

$$a_0^{kt}\sum_{i\in\mathcal{J}} x_{0ikt} \le s_{0kt} \le s_{J+1,kt} \le b_0^{kt}\sum_{i\in\mathcal{J}} x_{0ikt} \quad \forall k\in\mathcal{V}, \forall t\in\mathcal{T} \tag{18}$$

$$s_{0k,t+1} \ge R_{\mathrm{d}} - (1440 - s_{J+1,kt}) \quad \forall k\in\mathcal{V}, \forall t\in\mathcal{T} \tag{19}$$

$$\max_{t\in\mathcal{T}}\left(s_{0k,t+1} + \left(1440 - s_{J+1,k,t-1}\right) - R_{\mathrm{w}}\sum_{j\in\mathcal{J}} x_{0jk,t+1} - 2\cdot 1440\sum_{j\in\mathcal{J}} x_{0jkt}\right) \ge 0 \quad \forall k\in\mathcal{V} \tag{20}$$

$$x_{ijkt}\in\{0,1\} \quad \forall i,j\in\mathcal{J}_0\cup P, \forall k\in\mathcal{V}, \forall t\in\mathcal{T} \tag{21}$$

$$y_{kt}\in\{0,1\} \quad \forall k\in\mathcal{V}, \forall t\in\mathcal{T} \tag{22}$$

$$s_{ikt}\in\mathbb{N}\cup\{0\} \quad \forall i\in\mathcal{J}_0\cup P, \forall k\in\mathcal{V}, \forall t\in\mathcal{T} \tag{23}$$

The objective function (1) seeks to minimize the total working time of the nurses. Constraints (2) ensure that the clients are visited by the nurses who are allowed to treat them and who are available on the current day. Constraints (3) guarantee that each client is cared for on the days the visit is scheduled. Constraints (4) together with constraints (5) assure that the tours of the nurses start and end at the depot. If a nurse is required to take a break, the corresponding binary variable is set accordingly in constraints (6). For simplicity, we assume that breaks are taken directly at the homes of the clients. To model this assumption correctly and obtain feasible starting times, the predecessor and the successor of a break are the same client as stated in constraints (7). Constraints (8) ensure that the nurse is taking a break at a client that she/he is visiting. The nurses' working time per day is bounded by constraints (9) and constraints (10). The bounds depend on the break. In case a break is taken, the working time of the nurse is larger than or equal B and smaller than or equal L minutes. In case no break is taken, the working time needs to be smaller than B minutes. The nurses's working time per week is bounded by constraints (11). The starting times at clients who are visited in consecutive order are set appropriately in constraints (12). Constraints (13)–(14) guarantee correct starting times of the breaks. Additionally, constraints (16)–(17) ensure that the break is scheduled such that a consecutive working time of B is not exceeded. In constraints (15) the starting times at the clients are set within the given time windows, whereas in constraints (18) the same holds for the starting and finishing times of the daily tours. Constraints (19) ensure that the rest time between two consecutive working days of a nurse is at least R_{d}. Furthermore, constraints (20) guarantee that the nurses have at least once during the week, a sufficiently long rest time of $R_{\mathrm{w}} + 1440$. Constraints (21)–(23) state the feasible values for the decision variables.

4. BPC SOLUTION APPROACH

The problem formulation (1)–(23) is decomposed by Dantzig–Wolfe decomposition (Dantzig and Wolfe [12]). It results in a master problem and $V\cdot T$ subproblems. A solution of a subproblem corresponds to a feasible tour r for nurse k on day t with a working time of c_{kt}^r, which is equivalent to the sum of traveling and waiting times and service durations during the day. Breaks are not included in the working time. In the master problem, these tours are combined to form a feasible working plan for the week. Therefore, a variable λ_{kt}^r is associated with each solution of the subproblems. Further, binary variables ($\bar{\lambda}_{kt}$ and Λ_{it}) are introduced for the model formulation. $\bar{\lambda}_{kt}$ equals 0 if nurse k is working on day t and 1 otherwise. Λ_{it} equals 1 if client i is not visited on day t but has to be scheduled ($F_{it} = 1$) and 0 otherwise. Notice, that in a feasible solution to the original model formulation $\Lambda_{it} = 0, \forall i \in \mathcal{J}, \forall t \in \mathcal{T}$. However, as column generation is used to obtain the optimal tours, it might happen that some visits are unscheduled during the solution approach. Λ_{it} variables are, henceforth, introduced to prevent these cases. To obtain feasible solutions for the original problem, the costs of these variables equal a large constant M. Finally, the parameter f_{ir}^{kt} is given by the solution of the subproblem and equals 1 if client i is part of tour r by nurse k on day t and 0 otherwise (Table 2).

TABLE 2. Notations for the Branch-Price-and-Cut solution approach.

Notation	Definition
$\mathcal{R}^{kt}$	Set of feasible tours for nurse k on day t
c_{kt}^r	Working time of tour r by nurse k on day t (exclusive break length p in case a break is taken)
M	Large constant
f_{ir}^{kt}	Binary parameter, equals 1 if client i is part of tour r by nurse k on day t, 0 otherwise
$s_{0kt}^r, s_{J+1,kt}^r$	Starting and finishing time of tour r by nurse k on day t
λ_{kt}^r	Decision variable, is bigger than 0 if nurse k uses tour r on day t, 0 otherwise
Λ_{it}	Binary decision variable, equals 1 if client i is not visited on day t, 0 otherwise
$\bar{\lambda}_{kt}$	Binary decision variable, equals 1 if nurse k is not working on day t, 0 otherwise
α_{it}	Dual variable associated with constraints (25)
β_{kt}	Dual variable associated with constraints (26)
γ_k	Dual variable associated with constraints (27)

4.1. Master Problem

The mathematical formulation of the master problem is as follows:

$$\min \quad \sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{V}}\sum_{r\in\mathcal{R}^{kt}} c_{kt}^r\lambda_{kt}^r + \sum_{i\in\mathcal{J}}\sum_{t\in\mathcal{T}} M\Lambda_{it} \tag{24}$$

$$\text{s.t.} \quad \sum_{k\in\mathcal{V}}\sum_{r\in\mathcal{R}^{kt}} f_{ir}^{kt}\lambda_{kt}^r + \Lambda_{it} \geq F_{it} \quad \forall i \in \mathcal{J}, \forall t \in \mathcal{T} \tag{25}$$

$$\sum_{r\in\mathcal{R}^{kt}} \lambda_{kt}^r + \bar{\lambda}_{kt} = 1 \quad \forall k \in \mathcal{V}, \forall t \in \mathcal{T} \tag{26}$$

$$\sum_{t\in\mathcal{T}}\sum_{r\in\mathcal{R}^{kt}} c_{kt}^r\lambda_{kt}^r \leq W \quad \forall k \in \mathcal{V} \tag{27}$$

$$\sum_{t\in\mathcal{T}} \bar{\lambda}_{kt} \geq 1 \quad \forall k \in \mathcal{V} \tag{28}$$

$$R_{\mathrm{d}}\bar{\lambda}_{kt} + \sum_{r\in\mathcal{R}^{kt}} s_{0kt}^r\lambda_{kt}^r + \Big(1440 - \sum_{r\in\mathcal{R}^{k,t-1}} s_{J+1,k,t-1}^r\lambda_{k,t-1}^r\Big) \geq R_{\mathrm{d}} \quad \forall k \in \mathcal{V}, \forall t \in \mathcal{T} \tag{29}$$

$$\max_{t\in\mathcal{T}}\Big(\sum_{r\in\mathcal{R}^{k,t+1}} s_{0k,t+1}^r\lambda_{k,t+1}^r + \Big(1440 - \sum_{r\in\mathcal{R}^{k,t-1}} s_{J+1,k,t-1}^r\lambda_{k,t-1}^r\Big) + R_{\mathrm{w}}(\bar{\lambda}_{k,t+1} - 1) + 2\cdot 1440(\bar{\lambda}_{k,t} - 1)\Big) \geq 0 \quad \forall k \in \mathcal{V} \tag{30}$$

$$\lambda_{kt}^r \geq 0 \quad \forall k \in \mathcal{V}, \forall t \in \mathcal{T}, \forall r \in \mathcal{R}^{kt} \tag{31}$$

$$\bar{\lambda}_{kt} \in \{0,1\} \quad \forall k \in \mathcal{V}, \quad \forall t \in \mathcal{T} \tag{32}$$

$$\Lambda_{it} \in \{0,1\} \quad \forall i \in \mathcal{J}, \quad \forall t \in \mathcal{T} \tag{33}$$

The objective function (24) seeks to minimize the total working time of the nurses as in the original model formulation plus the costs of unscheduled visits. In constraints (25), it is guaranteed that each client is either part of a tour or unscheduled. Constraints (26) ensure that each nurse works on a certain day or rests. The maximum working time per week is given by constraints (27). Constraints (28) guarantee that each nurse has at least one day off. Sufficient rest times between two consecutive working days are respected by constraints (29), whereas sufficiently large rest times during the week are considered in constraints (30). Constraints (31)–(33) define the feasible values for the decision variables.

The number of different tours for each nurse and each day is exponentially large. Hence, we start solving the linear relaxation of this problem with an initial set of columns. This could be for instance the empty set, $\Lambda_{it} = 1, \forall i \in \mathcal{J}, \forall t \in \mathcal{T} : F_{it} = 1$, or the solution generated by the introduced VNS approach. After solving the master problem with an initial set of columns, the associated dual variables are used in the subproblem to compute new potential columns for the master problem. The master problem is resolved and this procedure repeats until no columns can be found anymore in the subproblem. In case the solution is noninteger branching is required. To keep the subproblem simple, we further relax constraints (29)–(30) from the master problem and treat these constraints in the branching scheme. As a consequence, the decision variables λ_{kt}^r can be set to integer values in the master problem.

4.2. Subproblem

A solution of the subproblem corresponds to a feasible tour of a nurse on a particular day. However, as there are a huge number of such tours, we are only interested in those tours that are leading to a better objective value. Therefore, we have to compute the reduced costs of the columns with the associated dual variables of constraints (25)–(27). Let the dual variables have the following valuation $\alpha_{it} \geq 0$, β_{kt} are unsigned, and $\gamma_k \leq 0$. The reduced cost of a column is then computed as follows:

$$c_{kt}^{r} - \sum_{i \in \mathcal{J}} f_{ir}^{kt} \alpha_{it} - \beta_{kt} - c_{kt}^{r} \gamma_k \tag{34}$$

The pricing problem for nurse k on day t can now be formulated in a similar way as the original problem:

$$\min \quad \sum_{i \in \mathcal{J}} \sum_{j \in \mathcal{J}} ((s_j - s_i)(1-\gamma_k) - \alpha_{it}) x_{ij} - (1-\gamma_k) y - \beta_{kt} \tag{35}$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{J}_0} x_{ij} = \sum_{h \in \mathcal{J}_0} x_{jh} \quad \forall j \in \mathcal{J} \tag{36}$$

$$\sum_{j \in \mathcal{J}} x_{0j} \leq 1 \tag{37}$$

$$\sum_{i \in \mathcal{J}} x_{iP} = y \tag{38}$$

$$x_{Pj} = x_{jP} \quad \forall j \in \mathcal{J} \tag{39}$$

$$s_{J+1} - s_0 \leq B + (L - B) y \tag{40}$$

$$s_{J+1} - s_0 \geq (B + p) y \tag{41}$$

$$s_i + (t_{ij} + d_i) x_{ij} \leq s_j + b_i (1 - x_{ij}) \quad \forall i,j \in \mathcal{J}_0 \tag{42}$$

$$s_P + p \cdot x_{Pj} \leq s_j + 1440(1 - x_{Pj}) \quad \forall j \in \mathcal{J} \tag{43}$$

$$s_i + (t_{ij} + d_i)(x_{ij} + x_{jP} - 1) \leq s_P + b_i (2 - x_{ij} - x_{jP}) \quad \forall i,j \in \mathcal{J} \tag{44}$$

$$s_P \leq s_0 + B \tag{45}$$

$$s_P \geq s_{J+1} - B \tag{46}$$

$$a_j \sum_{i \in \mathcal{J}_0} x_{ij} \leq s_j \leq b_j \sum_{i \in \mathcal{J}_0} x_{ij} \quad \forall j \in \mathcal{J} \tag{47}$$

$$a_0^{kt} \sum_{i \in \mathcal{J}} x_{0i} \leq s_0 \leq s_{J+1} \leq b_0^{kt} \sum_{i \in \mathcal{J}} x_{0i} \tag{48}$$

$$x_{ij} \in \{0,1\} \quad \forall i,j \in \mathcal{J}_0 \cup P \tag{49}$$

$$s_j \in \mathbb{N} \cup \{0\} \quad \forall j \in \mathcal{J}_0 \cup P \tag{50}$$

$$y \in \{0,1\} \tag{51}$$

The objective function (35) seeks to find tours with negative reduced costs as defined in (34). Constraints (36) guarantee that a nurse who is visiting a client is also leaving him again, whereas constraints (37) state that a nonempty tour starts at the depot. If a break must be taken in the tour, this is stated in constraints (38). Constraints (39) ensure that the break is taken directly at a client. The working time is bounded by constraints (40)–(41). The starting times at consecutively visited clients and breaks are set in constraints (42)–(44), whereas constraints (45)–(46) guarantee that the break is scheduled such that a consecutive working time of B is not exceeded. Constraints (47) ensure that the visits are scheduled within their time windows, whereas constraints (48) guarantee that the working time window of the nurse is respected. Finally, constraints (49)–(51) define the feasible values for the decision variables.

4.2.1. Labeling Algorithm.

This optimization problem is a shortest path problem from depot 0 to depot $J+1$ with the restrictions that a client is visited at most once along the path, the time windows are fulfilled, a break is scheduled if necessary, and the tour length of the path does not exceed the maximum working time. In literature, this problem is known as elementary shortest path problem with resource constraints. A characterization and different solution approaches can be found, for example, in Irnich and Desaulniers [20]. The problem is NP-hard (Dror [14]). A labeling algorithm based on dynamic programming is typically used to solve this problem. In the following, we introduce our labeling algorithm which is based on the work of Righini and Salani [30] and extended by concepts of Feillet et al. [17], Chabrier [10], Justesen and Rasmussen [21], Ceselli et al. [9], Bettinelli et al. [4], and our own ideas.

Label definition. In a labeling algorithm, a label is a path through the underlying network and associated with the following attributes:

- last client i visited on the path,
- set of visited clients $\mathcal{S}$ along the path,
- starting time s at client i,
- tour length τ from the start of the tour to client i,
- time buffer ρ,
- prize of the path δ,
- lower bound (LB) on costs χ,
- tour length before resp. after a break Υ, and
- rest time ϑ.

Due to time windows, the starting time of service s is never less than a_i and the tour length τ might include waiting times. However, as the starting time at the depot is not fixed, a postponement of the starting time of the tour can reduce current waiting times. Hence, ρ equals the remaining time buffer to postpone the visit at client i without violating any time windows along the path (see, e.g., Bettinelli et al. [4]).

Figure 1 shows the benefit of ρ. Along the path two clients i and j are visited. Traveling and service times are marked with bold lines, whereas waiting times are symbolized by dashed lines and the time buffer ρ by a dotted line. In the top scenario, it is assumed that the starting time at the depot is a_0^{kt}. Therefore, the nurse arrives too early at client j and has to wait. However, if the starting time at the depot can be postponed (bottom scenario), the waiting time can be reduced or even eliminated. As a consequence, also the tour length τ decreases.

δ equals the prize of the path, which is the sum of the collected dual variables α_{it} and β_{kt} along the path. As final labels are of interest only, in case of negative reduced cost, χ gives a LB on it (see, e.g., Justesen and Rasmussen [21]) by summing up the costs of attractive clients. A client i is attractive, if the service duration d_i minus the dual variable

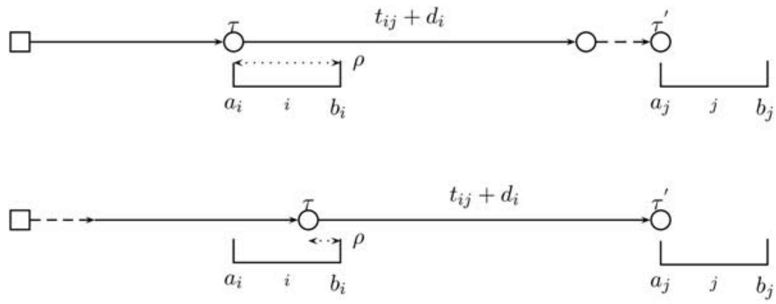

FIG. 1. Example for label extension (c.f., Bettinelli et al. [4]).

α_{it} is negative, because this leads to a reduction of the cost of the final label. Therefore, χ can be initialized by

$$\chi_0 = -\beta_{kt} + \sum_{i \in \mathcal{J}} \min(0, d_i - \alpha_{it}), \tag{52}$$

which is the maximum of negative reduced costs that can be gained in the subproblem. When extending the label to client j, χ is further updated by

$$\chi = \chi + \max(0, d_j - \alpha_{jt}). \tag{53}$$

Hence, if a label contains a client that is not attractive, the LB increases. In case $\chi \geq 0$ holds, the label can be fathomed, because no extension will lead to a final label with negative reduced costs.

As a label represents a tour of a nurse, we also have to guarantee that breaks are taken in case a consecutive working time of B minutes is exceeded. Hence, ϑ equals p in case a break is taken and Υ measures the tour length before resp. after a break. If Υ exceeds B minutes, a break needs to be inserted in the label. However, a break can also be scheduled before the tour length reaches B as long as the tour length after the break does not exceed B. This approach is useful to substitute waiting times. Figure 2 shows exemplary the result of inserting a break at client i in case waiting time arises at client j. In the top scenario waiting time arises when traveling from i to j. In the bottom scenario, a break is inserted and the waiting time can be eliminated.

The first label resides at the depot 0 with $S = \{0\}$ and no predecessor. The starting time equals $s = a_0^{kt}$. The tour length τ is set to 0. The initial prize equals half the dual variable of the nurse $\delta = \beta_{kt}/2$, the possible time slack is set to $\rho = \infty$, and the LB on the reduced cost is set to χ_0 as defined in (52). The tour length before resp. after a break Υ and the rest time ϑ are set to 0. A label is further represented by $\mathcal{L} = (i, \mathcal{S}, s, \tau, \rho, \delta, \chi, \Upsilon, \vartheta)$.

Label extension. A label $\mathcal{L} = (i, \mathcal{S}, s, \tau, \delta, \rho, \chi, \Upsilon, \vartheta)$ is extended to a label $\mathcal{L}' = (j, \mathcal{S}', s', \tau', \delta', \rho', \chi', \Upsilon', \vartheta')$ as follows with $\omega = \max(a_j - (s + t_{ij} + d_i), 0)$ and $\alpha_{it} = -\beta_{kt}$ for $i = 0$ and $i = J + 1$:

- $\mathcal{S}' = \mathcal{S} \cup \{j\}$
- $s' = \max(a_j, s + t_{ij} + d_i)$
- $\tau' = \tau + t_{ij} + d_i + \max(\omega - \rho, 0)$
- $\delta' = \delta + \alpha_{it}/2 + \alpha_{jt}/2$
- $\rho' = \min(\rho - \min(\rho, \omega), b_j - s')$
- $\chi' = \chi + \max(0, d_j - \alpha_{jt})$
- $\Upsilon' = \Upsilon + t_{ij} + d_i + \max(\omega - \rho, 0)$
- $\vartheta' = \vartheta$.

The service at client j cannot start before the beginning of its time window a_j nor before the nurse has arrived at client j after completing the service at the previous client i. If the nurse arrives early, the nurse has to wait for ω minutes to start service at a_j. However, in case $\rho > 0$, waiting time can be reduced by shifting the tour. If the waiting time can be eliminated, the new tour length of the path is computed by $\tau + t_{ij} + d_i$. Otherwise, the remaining waiting time $\omega - \rho$ has to be added to the tour length. ρ' is adapted accordingly. The LB on costs χ' increases if the associated dual variable is smaller than the service duration of client j. The update of the tour length before resp. after a break Υ' is the same as the update of the tour length τ' as long as Υ' does not exceed B and the rest time ϑ' is set to ϑ.

However, in case Υ' exceeds B minutes or the waiting time ω is bigger than p, a break is inserted. In this case, s, τ, ϑ, and Υ have to be computed with

- $s' = \max(a_j, s + t_{ij} + d_i + p)$
- $\tau' = \tau + t_{ij} + d_i + \max(\omega - \rho, p)$
- $\vartheta' = p$
- $\Upsilon' = 0$

If $0 < \omega < p$ holds, it might be beneficial to insert a break at this point rather than later in the tour to reduce waiting time. Therefore, an additional label is generated. For this label, the following extension rules hold:

- $\mathcal{S}' = \mathcal{S} \cup \{j\}$
- $s' = s + t_{ij} + d_i + p$
- $\tau' = \tau + t_{ij} + d_i + p$
- $\delta' = \delta + \alpha_{it}/2 + \alpha_{jt}/2$
- $\rho' = \min(\rho, b_j - s')$
- $\chi' = \chi + \max(0, d_j - \alpha_{jt})$
- $\vartheta' = p$
- $\Upsilon' = 0$

For both extension rules it holds that if either the tour length of the label exceeds the maximum working time, client j cannot be visited within its time window, or χ' is nonnegative, the label is discarded.

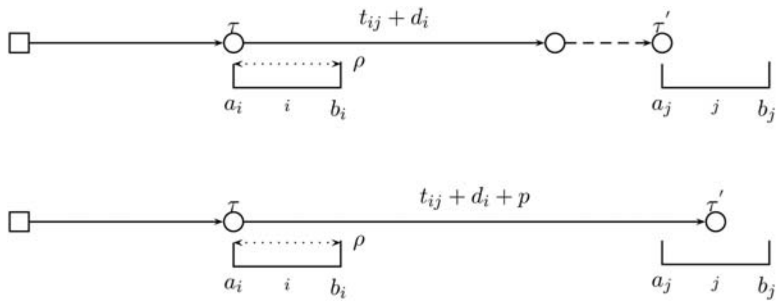

FIG. 2. Example for label extension with an insertion of a break.

Label dominance. Depending on the size of the graph and the tightness of the resource constraints a lot of labels might be generated that make the problem computationally intractable. However, the number of labels can be reduced by a dominance criterion. A label that is dominated by another label will never lead to a better final path, as the extension of the dominated label will always be worse or equal than the extension of a nondominated label. Therefore, dominated labels can be deleted.

A label $\mathcal{L}$ dominates another label $\mathcal{L}'$, if both are residing at the same client i and if all the following conditions (1–7) are met:

1. $\mathcal{S} \subseteq \mathcal{S}'$
2. $s \leq s'$
3. $\tau \leq \tau'$
4. $\tau - \vartheta \leq \tau' - \vartheta'$
5. $\rho \geq \rho'$
6. $\delta \geq \delta'$
7. $\Upsilon \leq \Upsilon'$

This means that all visited clients of $\mathcal{L}$ need to be visited by label $\mathcal{L}'$. The starting time, the tour length, and the current working time (the tour length minus the rest time) have to be smaller or equal. Furthermore, the possible time slack needs to be bigger or equal to reduce the total tour length. The prize of the label has to be bigger or equal and the tour length before resp. after a break has to be smaller or equal.

Extensions. Due to Condition 1, a lot of labels are not comparable. We have implemented an idea of Feillet et al. [17] who improve the efficiency of the algorithm by introducing the concept of unreachable clients. A client j is thereby defined as unreachable, if an extension of the current label residing at client i leads to a violation of the time window of client j or the maximum working time is exceeded. If set $\mathcal{S}$ is enlarged by these unreachable clients, more labels can be compared. As a consequence, the LB on costs is updated, as attractive clients might become unreachable. Suppose the path is extended from client i to client j and $\mathcal{F}$ is the set of unreachable clients at this point, the LB on costs χ' has to be updated as follows:

$$\chi' = \chi + \max(0, d_j - \alpha_{jt}) - \sum_{k \in \mathcal{F}} \min(0, d_k - \alpha_{kt}). \quad (54)$$

Moreover, a bounded bidirectional search as suggested by Righini and Salani [30] is used to improve the runtime. Thereby, we start with two initial labels residing at both depots. The forward label is $\mathcal{L}^{\mathrm{fw}} = (0, \{0\}, a_0^{kt}, 0, \beta_{kt}/2, \infty, \chi_0, 0, 0)$ starting at the starting depot. The backward label is $\mathcal{L}^{\mathrm{bw}} = (0, \{J+1\}, 0, 0, \beta_{kt}/2, \infty, \chi_0, 0, 0)$ starting at the finishing depot. The extension of the forward label follows the above stated rules. The extension of a backward label is symmetrical to the extension of a forward label. However, the time windows have to be modified, as the tour starts from the opposite direction. The latest arrival time at the depot equals the minimum of the end of the nurses' working time window b_0^{kt} and the latest arrival time at the depot when a client is visited before. The time windows are then computed as follows: $\hat{a}_i = \min(b_0^{kt}, \max_{j \in \mathcal{J}}(b_j + t_{j,0} + d_j)) - b_i - d_i$ and $\hat{b}_i = \min(b_0^{kt}, \max_{j \in \mathcal{J}}(b_j + t_{j,0} + d_j)) - a_i - d_i$. Additionally, Condition 3 of the dominance criteria has to be modified as proposed by Ceselli et al. [9] to find the optimal path(s). This is because breaks cannot be postponed if the consecutive working time exceeds B minutes. For the backward path, Condition 3 changes to $3'$ which is

$$3.' \tau + p \leq \tau'.$$

Both labels are extended as long as the resources stay within their bounds and the starting time at the current client does not exceed the half time of the possible working time window of the nurse, also called critical resource.

In a final step, forward labels $\mathcal{L}^{\mathrm{fw}}$ and backward labels $\mathcal{L}^{\mathrm{bw}}$ are joined to feasible paths. Thereby, we have to compute

$$\Upsilon^{\mathrm{fw}} + d_i + t_{ij} + d_j + \max(\omega, 0) + \Upsilon^{\mathrm{bw}} < B \quad \text{with} \quad (55)$$

$$\omega = \min(b_0^{kt}, \max_{i \in \mathcal{J}}(b_j + t_{j0} + d_j) - s^{\mathrm{bw}} - \rho^{\mathrm{bw}} - d_j - (s^{\mathrm{fw}} + \rho^{\mathrm{fw}} + d_i + t_{ij}). \quad (56)$$

In case (55) is violated, a break has to be inserted between the forward and the backward label. We define ξ to be 1 if a break needs to be inserted, and $\xi = 0$ otherwise. Additionally, the following criteria have to be fulfilled:

- $S^{\mathrm{fw}} \cap S^{\mathrm{bw}} = \emptyset$
- $s^{\mathrm{fw}} + d_i + t_{ij} + d_j + s^{\mathrm{bw}} + \xi p \leq b_0^{kt}$
- $\tau^{\mathrm{fw}} + d_i + t_{ij} + d_j + \max(\omega, \xi p) + \tau^{\mathrm{bw}} \leq L.$

If the joint label visits a client more than once, arrives too late at the depot, or exceeds the maximum tour length, the label is discarded. If the working time including the break exceeds B minutes, but is less than B minutes when subtracting the break length p the label is also discarded.

The reduced cost of the joined label depends on the working time of the path and is computed as:

$$(\tau^{\text{fw}} + d_i + t_{ij} + d_j + \max(\omega, \xi p) + \tau^{\text{bw}} - \vartheta)(1 - \gamma_k) - (\delta^{\text{fw}} + \delta^{\text{bw}} + \alpha_{it}/2 + \alpha_{jt}/2). \tag{57}$$

In case the working time exceeds B minutes, $\vartheta = p$, otherwise $\vartheta = 0$. If the cost of the label is nonnegative, the label is discarded.

Furthermore, in column generation, it is not required to find the best solution in each iteration, but rather one or several tours with negative reduced costs. Therefore, we first use a greedy algorithm as proposed by Bettinelli et al. [4] to find tours in a fast manner. This algorithm is based on a nearest neighbour concept. If no paths can be found by this algorithm, the exact algorithm will be used. To reduce the number of labels further, a heuristic dominance criterion as suggested by Chabrier [10] to improve the runtime, is applied. Hereby, only the number of visited clients is compared rather than the subset of visited clients. Therefore, Condition 1 is replaced by Condition 1′.

$$1.'|S| \leq |S'|$$

If no paths are found by the heuristic dominance criterion anymore, the algorithm switches to the exact dominance criterion.

Usually, all paths with negative reduced costs are added to the master problem, which is then resolved. However, in this case, the size of the master problem might become huge and the LP solver needs a long time to find a feasible solution. To keep the size of the master problem rather small, we do not add all paths with negative reduced costs to the master problem but only add a certain number of solutions.

In total, $V \cdot T$ subproblems have to be solved in each iteration of the column generation approach. However, this might take a lot of time. Hence, a partial pricing strategy based on a tabu list as proposed by Ceselli et al. [9] is used to improve the runtime. They suggest to solve each subproblem as long as new columns are found. In case no new columns are found for a particular subproblem, this subproblem is set on the tabu list for a certain number of iterations. If no new columns are found for any subproblem the tabu list is checked for new columns as well.

4.3. *Two-Path Inequalities*

To strengthen the LB two-path inequalities have been added. They have been introduced by Kohl et al. [23] for solving VRPs with time windows efficiently. The idea is quite simple. In the fractional solution, there might exist subsets of clients that are currently visited by less than two nurses but must be visited by at least two nurses. For instance, a nurse is currently assigned to two tours on a given day by a fractional value. However, in the integer solution, when combining these two tours into one, the maximum tour length or time windows would be violated. Hence, we have to solve a traveling salesman problem with time windows (TSPTW) and a maximum tour length to know if the subset can be visited by one nurse or not.

To identify these subsets, we have used the heuristic separation algorithm proposed by Kohl et al. [23]. The arising TSPTW is then solved with a dynamic programming algorithm similar to the presented labeling algorithm. In case the TSPTW is infeasible for subset $\mathcal{P}$ on day t, we add the following inequality to the master problem

$$\sum_{k \in \mathcal{V}} \sum_{r \in \mathcal{R}^{kt}} \mu_{r\mathcal{P}} \lambda_{kt}^r \geq 2, \tag{58}$$

with $\mu_{r\mathcal{P}}$ the number of arcs (i,j) used by tour r with client $i \in \mathcal{P}$ and $j \notin \mathcal{P}$.

As we use column generation, we search for additional inequalities at the end of each iteration when no new columns have been found. In case new inequalities are added to the master problem, the subproblems are resolved until no new columns can be found. Then again, we are using two-path inequalities to strengthen the LP formulation. In case inequalities are added, the supbroblems are resolved. This loop continues as long as new columns or inequalities are found.

The new inequalities influence the solution algorithm of the subproblem, as they are associated with dual variables that have to be considered during the pricing. As stated by Bettinelli et al. [4], we have to modify the reduced costs by subtracting $\mu_{r\mathcal{P}}\pi(\mathcal{P})$ with $\pi(\mathcal{P}) \geq 0$ being the associated dual variable to (58). This can, however, be translated to modified arc costs

$$\hat{t}_{ij} = t_{ij} - \sum\nolimits_{\mathcal{P}: i \in \mathcal{P}, j \in \mathcal{J} \setminus \mathcal{P}} \pi(\mathcal{P}). \tag{59}$$

Therefore, the structure of the subproblems remains the same and only the traveling times on the arcs have to be modified.

4.4. *Branching*

If neither columns nor cuts can be added to the master problem anymore and the solution includes fractional-valued variables, we use branching. Five different branching rules are applied to obtain integer solutions and feasible solutions with respect to the relaxed constraints (29)–(30).

4.4.1. Rule 1: A Client is Visited by Exactly One Nurse.

A typical branching decision to obtain integer solutions is branching on fractional values of the decision variables λ_{kt}^r. However, as columns are generated throughout the whole algorithm, columns fixed to 0 might be regenerated according to this strategy. Therefore, a constraint branching scheme is used. As each client has to be visited according to constraints

(25) and some decision variables have fractional value, it might occur that a client is visited by more than one nurse. We define v_{ikt} as follows:

$$v_{ikt} = \sum_{r \in \mathcal{R}^{kt}} f_{ir}^{kt} \lambda_{kt}^{r}. \tag{60}$$

If a client i is visited by nurse k on day t, $v_{ikt} > 0$. In case of fractionality, we branch on the v_{ikt} variables. The value closest to 1/2 is chosen for branching to have a balanced Branch-and-Bound tree. If v_{ikt} is set to 1 in the branch, client i is visited by nurse k on day t in the following. This means that this visit has to be scheduled in the corresponding subproblem.

As a consequence, all tours from other nurses $k' \neq k$ including client i on day t are deleted and client i is no longer included in the corresponding subproblems. Additionally, tours from nurse k without client i on day t are also deleted. In contrary, if v_{ikt} is set to 0 in the following branch, all tours of nurse k including client i on day t are deleted and client i is no longer included in the corresponding subproblem.

4.4.2. Rule 2: A Nurse Cannot Travel Partly Between Two Clients.

Additionally, it can happen that although each client is visited by only one nurse on a particular day the solution is still noninteger. In this case, the traveling of a nurse between two clients is stated by a noninteger value of x_{ij}. Therefore, we branch on the arcs x_{ij} and set $x_{ij} = 1$ in the left branch and $x_{ij} = 0$ in the right branch. Tours that do not fulfill these properties are deleted and the corresponding subproblem is modified accordingly.

Rules 3–5 state that each nurse needs sufficient rest times during the week. If the solution is infeasible due to constraints (29)–(30) branching is required as the daily rest times between two consecutive working days or the weekly rest times are not fulfilled. In case of noninteger solutions, there might exist several tours $\lambda_{kt}^{r} > 0$ for nurse k on day t. However, only setting some λ_{kt}^{r} to zero in case of infeasibility is not useful as these tours might be regenerated during the algorithm. Therefore, we branch on the working time windows of the nurses.

4.4.3. Rule 3: A Nurse Needs Sufficient Rest Time Between Two Consecutive Working Days.

The daily rest time between two consecutive working days has to be at least R_{d}. As several tours might exist for a nurse on a certain day, we compute the minimum starting time of all tours of a nurse on a certain day and the maximum finishing time of the tours of the same nurse on the previous day to check if violations occur. We compute the violations for all possible pairs (k, t) and take the highest value for branching. This is equivalent to evaluating

$$\max_{k \in \mathcal{V}, t \in \mathcal{T}} (R_{\mathrm{d}} - (\min_{r \in \mathcal{R}^{k,t+1}} s_{0k,t+1}^{r} + 1440 - \max_{r \in \mathcal{R}^{kt}} s_{J+1,k,t}^{r}), 0). \tag{61}$$

In case the highest violation occurs for nurse k from day t to day $t+1$, the working time windows of nurse k on

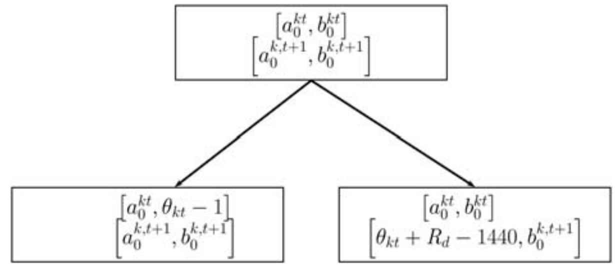

FIG. 3. Branching rule for daily rest time violations.

day t $\left[a_0^{kt}, b_0^{kt}\right]$ and day $t+1$ $\left[a_0^{k,t+1}, b_0^{k,t+1}\right]$ are divided. In the left branch, tours on day t have to finish before $\theta_{kt} = \max_{r \in \mathcal{R}^{kt}} s_{J+1,k,t}^{r}$. Hence, tours with a finishing time of θ_{kt} are deleted and the subproblem is solved with a new working time window. In case the daily rest time is still violated, further branching on the time windows is required. In the right branch, the starting time on day $t+1$ is modified to $\theta_{kt} + R_{\mathrm{d}} - 1440$ to fulfill the daily rest time between day t and $t+1$ in case nurse k starts her tour at a_0^{kt} on day t. Tours starting earlier than the new starting time are deleted. Again, it might happen that the daily rest times are still violated by the new solution. In this case, further branching on the working time windows is required. Figure 3 shows exemplary the working time windows in each branch.

4.4.4. Rule 4: Each Nurse Needs at Least One Day Off Per Week.

According to the weekly rest time restrictions, we have to assure that each nurse has at least one day off during the week. If this is not the case, several branches are generated. Each branch corresponds to a possible day off. Hence, in each branch the working time window of the day is set to an empty time window and $\overline{\lambda}_{kt} = 1$. As the planning horizon corresponds to one week, at most seven branches are used. Notice, however, that we might have less than seven branches, because a nurse might be forced to visit a client on a certain day due to Rule 1 or 2. In that case, this day is not considered in the branching. Figure 4 shows the possible seven branches. All existing tours from a nurse on the corresponding branching day are deleted henceforth and no subproblem has to be solved for this nurse on the particular day.

4.4.5. Rule 5: A Nurse Needs a Sufficiently Large Weekly Rest Time.

In case the weekly rest times are still not fulfilled using Rule 4, we additionally branch on the working time windows of day $t-1$ and day $t+1$, if day t is a day off. Therefore,

$$\min_{r \in \mathcal{R}^{k,t+1}} s_{0k,t+1}^{r} + 1440 - \max_{r \in \mathcal{R}^{k,t-1}} s_{J+1,k,t-1}^{r} \tag{62}$$

is computed for all days off ($\overline{\lambda}_{kt} = 1$) of nurse k and we branch on the pair (k, t) with the highest violation of the weekly rest time constraint. If a nurse has several days off, and the weekly rest time is still violated, the number of branches

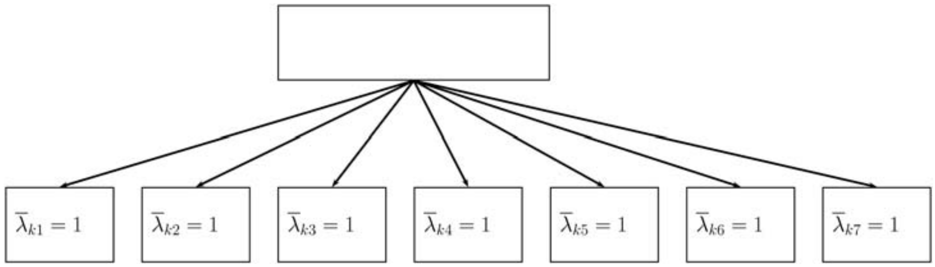

FIG. 4. Branching rule for a day off per week.

equals the number of days off times two. Again, the working time windows are divided such that in the left branch tours are forbidden to finish at time $\theta_{k,t-1} = \max_{r \in \mathcal{R}^{k,t-1}} s^r_{J+1,k,t-1}$ on day $t-1$, whereas in the right branch tours are not allowed to start earlier than $\theta_{k,t-1} + R_w - 1440$ on day $t+1$. All tours that are exceeding these bounds are deleted. Figure 5 shows exemplary the branching scheme.

We have used the following order of the branching rules: Rules 4, 1, 5, 3, and 2. Rule 4 is identified for having the most impact, because tours for a nurse on a certain day are forbidden. Additionally, in case Rule 1 or 2 is used before, we might end up with solutions in which a nurse is scheduled on each single day of the week, which is infeasible. However, as Rules 3 and 5 might only influence the working time windows slightly and a lot of further branching would be necessary, Rule 1 gets a higher priority. Then, follow Rules 5 and 3 to guarantee enough rest times. Only in case the solutions are still containing fractional-valued variables, Rule 2 is needed. A best-first search is used as tree search strategy.

4.5. *Initial Solution*

Basically, we can just start the BPC algorithm with the "most" infeasible solution in which all clients are not visited ($\Lambda_{it} = 1, \forall i \in \mathcal{J}, \forall t \in \mathcal{T} : F_{it} = 1$). However, a good starting solution might prune a lot of branches in the Branch-and-Bound tree. A VNS-based solution approach is used to start the BPC algorithm with good quality solutions. The algorithm was developed by us and implemented in C++.

VNS is a metaheuristic solution approach and used for various VRP applications (see, e.g., Hansen and Mladenović [19]). Metaheuristics are usually fast, but sometimes they tend to be trapped in local optima. VNS tries to overcome these local optima by combining a local search procedure that finds these optima with a shaking procedure in which a random solution of the neighborhood of this solution is picked. In case no better solution is found by local search, the neighborhood is enlarged.

In Figure 6, the complete procedure of the VNS algorithm is summarized. In the initialization step, the clients are added to a nurse with the appropriate skills and ordered according to their time windows. Two operators are used in the shaking phase. In the smaller neighborhoods, a move operator is used to modify the solution. Thereby, a sequence of clients is moved from one nurse to another nurse who is allowed to service these clients. In the larger neighborhoods, a cross-exchange operator is applied. In this case, two sequences of clients are exchanged between nurses who are allowed to service these clients. In the local search procedure, the new tours are optimized with an adapted 2-opt procedure according to Savelsbergh [31]. This version takes the forward time slack into account to minimize the total working time. The breaks are inserted afterward. The new solution is evaluated according to a penalized weighted objective function. The acceptance criterion is hereby based on this function and a probability function adapted from simulated annealing (see, e.g., Kirkpatrick et al. [22]). If the new solution is accepted, it becomes the new incumbent and the neighborhood structure κ is reset to the first one. If it also is feasible and better than the previous best found solution (BFS), it is

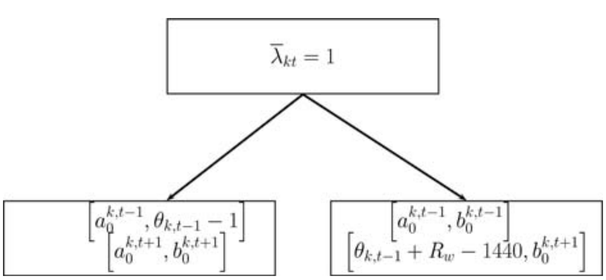

FIG. 5. Branching rule in case of weekly rest time violations.

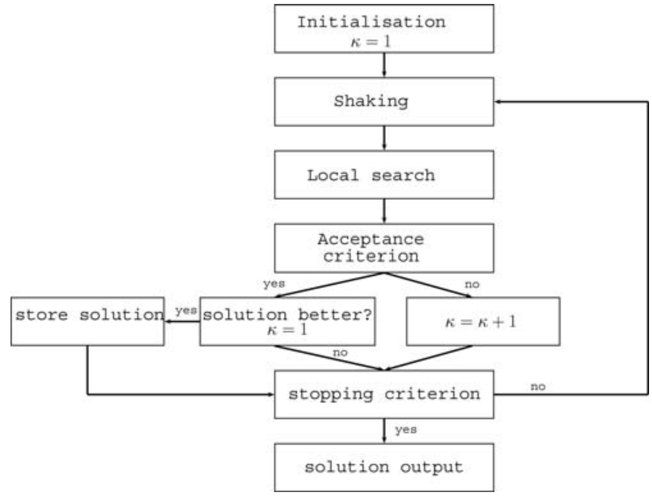

FIG. 6. Variable Neighborhood Search solution approach.

stored. In case it is not accepted, the neighborhood structure is changed. Finally, the stopping criterion is checked. If the number of iterations is smaller than the given threshold, the algorithm starts over with the shaking procedure. Otherwise, the algorithm terminates.

5. COMPUTATIONAL RESULTS

Planning HHC services in a medium-term context is quite new in research. In literature, this problem has been rarely addressed (Gamst and Jensen [18], Nickel et al. [25], Cattafi et al. [8]) and if so, with different assumptions and constraints. Hence, useful benchmarking instances are missing.

We have generated the test instances in accordance with real-life conditions at the Austrian Red Cross (ARC), one of the largest HHC providers in Austria. This organization employs nurses with three different qualification levels. A nurse with a qualification level of 1 can aid in the household, like cleaning, whereas a nurse with a qualification level of 3 does more advanced HHC, like giving injections. In the test instances, the qualification levels of the nurses and the required qualifications at the clients are equally distributed. Whether a client (resp. a nurse) is speaking a language or not is set randomly, with both options equally likely. It only has to be guaranteed that a client (resp. a nurse) speaks at least one language. In total, we assume four languages. Rejections of clients (resp. nurses) are also set randomly, but such that a feasible solution can be obtained. The duration of the service of a client is normally distributed with a mean dependent on the qualification level of the service and a standard deviation of 15 min. The minimum duration is set to 5 min. The means for the durations were collected empirically from data provided by the ARC. They are 46.69 min for a qualification level of 1, 39.71 min for a qualification level of 2, and 28.18 min for a qualification level of 3. The durations are rounded to integer values. Each working day is partitioned into six time windows with a duration of 2 h in which service might occur. The time windows are equally distributed. The frequency of the clients' visits is set randomly between one and seven with each day equally likely to be a visiting day. The nurses are available on each day during the week. The maximum working time per week is set to $W = 2,400$ min, whereas the maximum working time per day equals $L = 630$ min. The maximum consecutive working time without a break is set to $B = 360$ min with a break length of $p = 30$ min. The daily rest time equals $R_\mathrm{d} = 660$ min, and the parameter for the weekly rest time is $R_\mathrm{w} = 720$ min. The clients, the nurses, and the depot are equally distributed on an area of 50×50. Euclidean distances are used for computing the traveling times. The maximum traveling time equals 70 min.

For the computational experiments, two different scenarios have been used for each test instance. In the first scenario ($D = 0$), no downgrading of qualification levels is allowed. Hence, if a nurse wants to visit a client, the qualification levels of the nurse and the client have to be the same. In the second scenario ($D = 1$), downgrading is allowed but only from one level to the next lower level. Therefore, a nurse can also service a client, if her qualification level is one level higher than the qualification level required by the client. However, a nurse with a qualification level of 3 is not allowed to visit a client who requires a qualification level of one. This is common in practice as the nurses as well as the company are satisfied when the employees are more or less deployed according to their qualification skills.

The original problem formulation was implemented with the solver software Xpress MP 7.3. However, the results show that only small test instances including two nurses and 10 clients can be solved to optimality. Bigger instances are not solvable.

The BPC algorithm was implemented using the COIN-BCP framework Version 1.3.4 (Lougee-Heimer [24]). All tests were run on an Intel 2.80GHz Core i7 CPU with 4 cores and 6 GB RAM with Linux as operating system. The algorithm was run on a single core. However, Cplex 12.4, which was chosen as LP solver, used the remaining cores. For solving the subproblem, the framework "r_c_shortest_path.hpp" provided by the Boost Libraries C++, Version 1.46.1 (Siek et al. [32]) served as a basis. It was further modified according to the concepts introduced in subsection 4.2.1.

In the following, we present the results of the BPC solution approach starting with an empty solution or using the VNS solution as initial solution. In the latter case, columns are generated from the VNS solution and added to the problem. The results, obtained after a maximum runtime of 1 h, are summarized in Table 3. For each test instance the number of nurses (V), the number of clients (C), and the total number of visits during the week ($\mathbf{V}$) are shown. Additionally, D informs whether downgrading of qualification levels is allowed (1) or not (0). In case the VNS algorithm is used, its solution after 10^6 iterations is reported. Besides, the LB obtained in the root node, the best found lower bound (BLB), the BFS as well as the gap between the BLB and the BFS are shown. Then, the number of nodes, the depth of the Branch-and-Bound tree, and the number of generated variables as well as cuts are given. Moreover, as different heuristics are used to solve the subproblems, the number of subproblems solved by each method as well as the number of times solutions were obtained using these methods are stated. Thereby, greedy relates to the greedy algorithm, heur stands for the labeling algorithm using the heuristic dominance criterion, and exact stands for the exact labeling algorithm. Finally, the total runtime that is the sum of the runtime of the BPC approach and the computation time for the initial solution using VNS is shown. If the algorithm does not terminate within 1 h, the runtime is marked with an asterisk and the best obtained solution only serves as an upper bound.

For test instances 1–3, results are only presented in case downgrading is allowed. Otherwise, no feasible solutions that include all visits can be obtained. This is because in test instances 1 and 2 no nurse with a qualification level of 1 is present, although some clients do require a qualification level of 1. In test instance, three nurses for each qualification level are available. However, it is not possible for one nurse

TABLE 3. Results for the Branch-Price-and-Cut approach.

					VNS	LB in			Gap					Number of subproblems			Solutions obtained			Sol. time
	V	J	**V**	D	Sol.	Root node	BLB	BFS	BLB/BFS	Nodes	Depth	Vars	Cuts	Greedy	Heur	Exact	Greedy	Heur	Exact	(s)
1	2	10	32	1	—	2,860.00	2,973.00	2,973	0.00%	8	1	587	5	366	287	240	79	47	0	0.14
1	2	10	32	1	2,973	2,860.00	2,973.00	2,973	0.00%	8	1	613	5	386	306	253	80	53	0	6.70
2	3	15	59	1	—	4,212.00	4,216.00	4,216	0.00%	5	2	1,392	3	407	316	277	91	39	0	0.21
2	3	15	59	1	4,284	4,212.00	4,216.00	4,216	0.00%	7	2	796	3	507	412	360	95	52	0	10.06
3	4	20	76	1	—	4,988.80	5,000.00	5,000	0.00%	89	8	12,261	17	6,999	6,338	5,961	661	377	0	7.54
3	4	20	76	1	5,043	4,988.80	5,000.00	5,000	0.00%	32	6	3,390	18	2,628	2,450	2,259	178	191	0	14.01
4	6	30	100	0	—	0.00	8,778.00	8,778	0.00%	1	0	751	0	233	150	122	83	28	0	0.03
4	6	30	100	0	8,783	0.00	8,778.00	8,778	0.00%	1	0	284	0	227	139	115	88	24	0	9.20
4	6	30	100	1	—	6,920.50	6,978.00	6,978	0.00%	43	3	4,140	2	4,259	3,973	3,710	286	263	0	3.76
4	6	30	100	1	7,005	6,920.50	6,978.00	6,978	0.00%	15	2	3,058	2	2,160	1,847	1,634	313	213	0	12.73
5	7	35	122	0	—	10,087.00	10,089.00	10,089	0.00%	3	1	1,045	2	756	576	510	180	66	0	0.19
5	7	35	122	0	10,112	10,087.00	10,089.00	10,089	0.00%	3	1	884	2	782	562	477	220	85	0	11.62
5	7	35	122	1	—	7,450.57	7,462.00	7,462	0.00%	63	5	13,208	2	4,867	4,552	4,251	315	301	0	35.24
5	7	35	122	1	7,545	7,450.57	7,462.00	7,462	0.00%	47	4	15,088	2	5,115	4,605	4,184	510	421	1	52.47
6	8	40	153	0	—	9,335.00	9,337.00	9,337	0.00%	5	2	2,736	0	1,320	1,052	963	268	89	0	0.53
6	8	40	153	0	9,351	9,335.00	9,337.00	9,337	0.00%	5	2	1,791	0	1,214	907	777	307	130	0	12.70
6	8	40	153	1	—	7,821.00	7,858.00	7,858	0.00%	876	29	259,114	119	146,683	142,251	135,441	4,432	6,810	47	3,600.14
6	8	40	153	1	7,988	7,821.00	7,858.00	7,858	0.00%	228	7	32,395	12	34,033	33,159	31,130	874	2,029	19	909.58
7	9	45	177	0	—	12,147.67	12,338.00	12,338	0.00%	104	13	11,594	14	20,493	19,591	18,693	902	898	13	29.12
7	9	45	177	0	12,628	12,147.67	12,338.00	12,338	0.00%	107	9	8,207	16	14,472	13,388	12,288	1,084	1,100	0	24.91
7	9	45	177	1	—	10,512.33	10,578.60	10,614	0.33%	1,360	10	102,621	410	157,545	155,983	149,801	1,562	6,182	69	3,600.11*
7	9	45	177	1	10,901	10,512.19	10,512.19	10,828	3.00%	1,130	23	1,382,827	192	215,882	200,616	187,802	15,266	12,814	122	3,600.82*
8	10	50	218	0	—	14,075.59	14,084.00	14,084	0.00%	186	6	23,672	54	31,179	30,074	28,255	1,105	1,819	5	209.26
8	10	50	218	0	14,523	14,075.59	14,084.00	14,084	0.00%	613	29	107,625	142	203,579	198,519	191,443	5,060	7,076	8	1,337.38
8	10	50	218	1	—	12,123.38	12,123.38	12,489	3.02%	89	13	45,935	16	15,892	14,968	14,145	924	823	22	3,608.09*
8	10	50	218	1	12,724	12,123.37	12,142.29	12,724	4.79%	85	3	54,388	25	10,903	10,269	9,422	634	847	28	3,616.40*
9	12	60	255	0	—	14,960.86	15,021.69	—	No UB	2,644	13	126,729	944	379,291	373,966	363,853	5,325	10,113	574	3,600.50*
9	12	60	255	0	15,374	14,960.86	15,021.69	15,374	2.35%	2,540	14	124,333	887	388,375	383,144	372,836	5,231	10,308	370	3,600.22*
9	12	60	255	1	—	13,189.76	13,189.76	—	No UB	22	2	52,872	19	4,292	3,768	3,461	524	307	12	3,663.76*
9	12	60	255	1	13,808	13,189.76	13,189.76	13,784	4.51%	15	2	70,084	16	4,081	3,502	3,173	579	329	16	3,627.60*

TABLE 4. Results for 10 additional instances including six nurses and 30 clients.

	V	J	$\mathbf{V}$	D	VNS Sol.	LB in Root node	BLB	BFS	Gap BLB/BFS	Nodes	Depth	Vars	Cuts	Number of subproblems Greedy	Heur	Exact	Solutions obtained Greedy	Heur	Exact	Sol. time (s)
4a	6	30	114	1	—	7,026.50	7,029.00	7,029	0.00%	5	2	6,560	1	922	745	651	177	94	0	1.86
4a	6	30	114	1	7,191	7,026.50	7,029.00	7,029	0.00%	3	1	1,851	1	764	586	487	178	99	3	15.47
4b	6	30	104	1	—	7,228.94	7,261.00	7,261	0.00%	1,315	48	292,407	0	176,882	148,728	138,877	28,154	9,851	0	136.86
4b	6	30	104	1	7,283	7,228.94	7,261.00	7,261	0.00%	380	13	19,986	0	37,184	34,383	32,845	2,801	1,538	0	33.07
4c	6	30	121	1	—	0.00	7,338.00	7,338	0.00%	1	0	3,703	0	400	245	196	155	49	1	0.49
4c	6	30	121	1	7,431	0.00	7,338.00	7,338	0.00%	1	0	1,951	0	580	454	396	126	58	0	14.65
4d	6	30	130	1	—	8,313.05	8,375.00	8,375	0.00%	1,018	33	329,552	77	188,438	175,561	166,203	12,877	9,358	57	946.67
4d	6	30	130	1	8,458	8,313.05	8,375.00	8,375	0.00%	265	34	121,108	17	52,034	48,729	45,906	3,305	2,823	8	291.37
4e	6	30	104	1	—	6,547.50	6,558.00	6,558	0.00%	12	2	3,381	2	1,913	1,637	1,486	276	151	2	1.46
4e	6	30	104	1	6,578	6,547.50	6,558.00	6,558	0.00%	12	2	2,862	2	2,104	1,851	1,638	253	213	4	13.23
4f	6	30	124	1	—	8,752.36	8,762.00	8,762	0.00%	27	7	10,543	2	5,994	5,585	5,208	409	377	0	30.66
4f	6	30	124	1	8,987	8,752.36	8,762.00	8,762	0.00%	84	9	13,869	16	11,362	10,630	9,879	732	751	0	1,291.47
4g	6	30	122	1	—	8,126.99	8,233.00	8,233	0.00%	200	6	16,763	18	21,280	20,430	19,136	850	1,294	0	79.94
4g	6	30	122	1	8,247	8,119.20	8,233.00	8,233	0.00%	114	9	25,400	19	28,200	27,049	25,364	1,151	1,685	0	82.95
4h	6	30	105	1	—	6,377.10	6,392.00	6,392	0.00%	868	35	143,560	56	100,212	94,402	89,792	5,810	4,610	14	334.98
4h	6	30	105	1	6,544	6,376.00	6,392.00	6,392	0.00%	217	9	36,897	20	21,338	20,676	19,259	662	1,417	3	89.10
4i	6	30	120	1	—	7,515.16	7,556.00	7,556	0.00%	272	28	25,870	20	30,188	28,302	26,678	1,886	1,624	18	96.29
4i	6	30	120	1	7,658	7,515.16	7,556.00	7,556	0.00%	191	28	22,027	33	26,688	24,961	23,513	1,727	1,448	19	115.43
4j	6	30	98	1	—	6,398.70	6,408.00	6,408	0.00%	14	3	2,383	2	1,929	1,760	1,639	169	121	1	0.99
4j	6	30	98	1	6,515	6,398.70	6,408.00	6,408	0.00%	5	2	1,855	2	1,014	815	724	199	91	0	11.11

TABLE 5. Results for 10 additional instances including nine nurses and 45 clients.

	V	C	**V**	D	VNS Sol.	LB in Root node	BLB BLB	BFS BFS	Gap BLB/BFS	Nodes	Depth	Vars	Cuts	Number of subproblems Greedy	Heur	Exact	Solutions obtained Greedy	Heur	Exact	Sol. time (s)
7a	9	45	173	0	—	10,810.57	10,823.00	10,823	0.00%	167	8	18,216	11	27,526	26,443	24,938	1,083	1,505	0	73.46
7a	9	45	173	0	10,980	10,810.57	10,823.00	10,823	0.00%	161	8	22,526	11	26,992	25,850	24,317	1,142	1,533	2	86.05
7a	9	45	173	1	—	9,679.70	9,686.10	—	No UB	491	6	67,318	127	18,107	17,713	16,634	394	1,079	22	3,609.23*
7a	9	45	173	1	9,995	9,679.70	9,679.70	9,840	1.66%	172	23	297,932	40	24,960	23,075	21,253	1,885	1,822	30	3,603.25*
7b	9	45	160	0	—	10,896.30	10,916.00	10,916	0.00%	32	4	3,821	6	5,756	5,306	5,004	450	302	1	4.34
7b	9	45	160	0	11,006	10,896.30	10,916.00	10,916	0.00%	24	4	3,717	4	4,799	4,300	3,993	499	307	1	15.28
7b	9	45	160	1	—	9,765.50	9,795.00	9,795	0.00%	524	8	39,932	26	50,176	48,867	45,707	1,309	3,160	52	735.89
7b	9	45	160	1	9,984	9,765.50	9,765.50	9,833	0.69%	1,159	20	324,678	23	268,828	261,211	249,164	7,617	12,047	546	3,600.38*
7c	9	45	194	0	—	14,693.22	14,693.22	—	No UB	5,255	59	1,108,487	239	1,278,172	1,146,088	1,085,394	132,084	60,694	26	3,600.06*
7c	9	45	194	0	15,297	14,675.47	15,054.00	15,054	0.00%	4,919	30	240,723	194	761,620	724,551	694,733	37,069	29,818	94	2,191.42
7c	9	45	194	1	—	11,714.92	11,714.92	—	No UB	1,620	708	507,393	25	198,262	184,409	175,196	13,853	9,213	373	3600.76*
7c	9	45	194	1	12,276	11,714.88	11,714.88	12,271	4.75%	1,180	555	327,288	23	165,843	154,092	147,082	11,751	7,010	26	3,600.04*
7d	9	45	203	0	—	12,620.38	12,620.38	12,844	1.77%	2,550	28	631,954	605	666,828	636,377	611,104	30,491	25,233	46	3,600.04*
7d	9	45	203	0	12,966	12,620.38	12,690.00	12,690	0.00%	1,293	23	134,480	275	190,673	182,617	175,428	8,056	7,189	16	1,181.77
7d	9	45	203	1	—	10,841.71	10,841.71	—	No UB	57	16	66,777	27	6,153	5,741	5,342	412	399	5	3,630.38*
7d	9	45	203	1	11,209	10,841.71	10.868.22	11,209	3.14%	57	2	64,870	31	6,299	5,783	5,257	516	526	8	3,613.27*
7e	9	45	199	0	—	12,292.00	12,302.00	12,302	0.00%	5	2	6,895	6	1,667	1,358	1,177	309	181	2	2.82
7e	9	45	199	0	12,314	12,292.00	12,302.00	12,302	0.00%	5	2	3,990	8	1,830	1,427	1,177	403	250	2	16.93
7e	9	45	199	1	—	10,923.83	10,944.71	—	No UB	326	8	40,556	64	31,287	30,843	29,410	444	1,433	59	3,601.10*
7e	9	45	199	1	11,206	10,923.83	10,923.83	11,138	1.96%	195	51	450,257	17	77,240	71,039	65,434	6,201	5,605	75	3,600.14*
7f	9	45	193	0	—	13,066.70	13,133.00	13,133	0.00%	341	16	34,069	103	59,788	57,800	55,133	1,988	2,667	0	231.14
7f	9	45	193	0	13,811	13,066.70	13,133.00	13,133	0.00%	304	16	35,804	49	58,229	56,197	53,489	2,032	2,708	1	210.31
7f	9	45	193	1	—	10,885.74	10,918.81	—	No UB	661	12	65,216	100	51,196	49,878	46,888	1,318	2,990	28	3,601.72*
7f	9	45	193	1	11,206	10,885.74	10,885.74	11,174	2.65%	15	2	19,972	14	1,814	1,291	1,102	523	189	2	3,706.65*
7g	9	45	165	0	—	10,855.30	10,955.00	10,955	0.00%	98	7	6,583	23	12,187	11,585	10,968	602	617	4	12.88
7g	9	45	165	0	11,088	10,855.30	10,955.00	10,955	0.00%	68	7	6,664	23	11,137	10,469	9,805	668	664	7	25.01
7g	9	45	165	1	—	9,145.61	8,176.06	—	No UB	911	11	77,837	207	124,160	123,134	118,935	1,026	4,199	97	3,600.14*
7g	9	45	165	1	9,323	9,145.61	9,145.61	9,293	1.61%	433	15	103,140	84	140,422	137,558	132,348	2,864	5,210	52	3,600.06*
7h	9	45	172	0	—	12,857.00	12,937.00	12,937	0.00%	37	5	7,649	15	6,940	6,503	6,224	437	279	0	7.60
7h	9	45	172	0	13,318	12,857.00	12,937.00	12,937	0.00%	37	5	4,896	15	7,186	6,758	6,444	428	314	0	20.52
7h	9	45	172	1	—	10,706.48	10,749.15	10,751	0.02%	1,222	13	81,244	311	140,894	137,490	132,855	3,404	4635	211	3600,64*
7h	9	45	172	1	11,140	10,706.48	10,749.00	10,751	0.02%	1,203	13	89,230	301	144,985	141,266	136,033	3,719	5233	162	3601,31*
7i	9	45	161	0	—	10,396.00	10,437.00	10,437	0.00%	189	7	14,966	11	29,028	27,777	26,670	1,251	1,107	0	36.59
7i	9	45	161	0	10,485	10,396.00	10,437.00	10,437	0.00%	129	7	12,222	11	20,141	18,834	17,963	1,307	871	2	34.19
7i	9	45	161	1	—	8,955.92	8,999.00	8,999	0.00%	613	7	57,195	41	60,581	59,310	55,073	1,271	4,237	60	2050,14
7i	9	45	161	1	9,181	8,955.92	8,955.92	9,080	1.39%	748	20	161,698	19	119,459	111,895	106,635	7,564	5260	18	3600,99*
7j	9	45	183	0	—	10,974.38	11,129.00	11,129	0.00%	474	16	52,471	20	55,884	52,696	49,038	3,188	3,658	3	142.95
7j	9	45	183	0	11,348	10,974.38	11,129.00	11,129	0.00%	276	16	54,748	24	44,015	40,705	37,530	3,310	3,175	32	104.14
7j	9	45	183	1	—	9,153.00	9,159.00	9,159	0.00%	141	6	46,863	19	31,300	30,624	28,569	676	2,055	52	895.44
7j	9	45	183	1	9,483	9,153.00	9,159.00	9,159	0.00%	141	6	44,542	19	34,352	33,588	31,814	764	1,774	42	950.06

in this test instance to perform five visits in the course of a day, because of the rest time constraints.

Generally, it can be observed that the scenario with no downgrading always leads to worse solutions compared with the scenario with downgrading. The advantage of no downgrading is faster runtimes. However, this conclusion is not surprising as a subproblem then only consists of a nurse and clients with the same qualification level. The sizes of the subproblems become smaller and the solution times of the subproblems decrease, which leads to smaller total runtimes.

The results of Table 3 show that small test instances can be solved within short computation time for both setups. Additionally, when using the VNS solution as an initial upper bound, the generated Branch-and-Bound tree is typically smaller. However, the total runtime is often worse when using the VNS solution. This is because the total runtime includes the runtime for the VNS approach. The computation of the VNS solution for 10^6 iterations took less than 27s for the biggest presented test instances.

For the bigger test instances, it can be observed that in case the global optimal solution is not obtained within 1 h of computation time, the gap between the BFS and the best LB is rather small.

The size of the test instances is further shortened as follows: V/J. Test instances with up to a size of 9/45 and 177 visits during the week with no downgrading have been solved fast. Problem instances above this size need more runtime for finding optimal solutions or proving the optimality of the best found upper bounds. Additionally, we generated 10 test instances with 6/30 to test if these instances can be solved to optimality without any difficulty. These results are presented in Table 4. Moreover, another 10 test instances with a size of 9/45 have been generated to test whether these can be solved to optimality within 1 h too, or if this problem size is the limit of the presented BPC solution approach. These results are shown in Table 5.

The results of Table 4 show that all test instances with a size of 6/30 with the number of visits varying from 98–130 during the week are solved to optimality within 1, 245 s of runtime. However, the following observations can be made: In case the VNS solution is in a bad area of the solution space as in 4f, the BPC algorithm is much faster when used without this solution. However, if the VNS solution is in the right part of the solution space (e.g., 4b,4d,4h) the runtime of the algorithm decreases significantly because the Branch-and-Bound tree is much smaller.

The test instances with a size of 9/45 are rather difficult to solve. From the 10 new instances, we were able to solve all to optimality within the given time limit for the case that $D = 0$ and the algorithm is started with the VNS solution. In case we start from scratch, only eight test instances could be solved to optimality. Therefore, it is beneficial to start the exact algorithm with a good upper bound. In case $D = 1$, only three instances could be solved to optimality, and one only if the VNS algorithm has been used as upper bound. Generally, it seems that for these instances the solution space is too big to find the optimal solutions within 1 h of computing time and some VNS solutions lie in a bad area of the solution space. Conversely, it is worth mentioning that good feasible solutions have been found for all test instances with $D = 1$ when the VNS was used for the initial solutions, whereas no feasible solutions have been found for six test instances when the algorithm was started from scratch.

Additonally, it can be remarked that the heuristic algorithms used for solving the subproblems work well. The greedy algorithm and the labeling algorithm with the heuristic dominance criterion were often able to find new solutions for the subproblems. However, in many cases the exact algorithm was also used but only found new solutions rarely. This is a good indicator that the heuristics for labeling work well.

6. CONCLUSIONS

Medium-term HHC planning consists of several decisions that should be made simultaneously rather than sequentially to obtain solutions of good quality. Besides, different assignment constraints, the clients have to be visited within their time windows. Additionally, working time restrictions such as maximum working time per day, maximum working time per week, and breaks during the days have to be considered. Furthermore, rest times between consecutive working days as well as weekly rest times have to be planned. In literature, the periodic aspect of HHC planning is rather unexplored. However, the few papers that deal with this planning horizon do not incorporate the different working time restrictions and types of required breaks.

We propose a mathematical model formulation and tried to solve it with standard solver software. As only very small problem instances were solvable, we developed a BPC solution algorithm to obtain global optimal solutions. Additionally, a VNS-based solution approach is used to provide good initial upper bounds for the BPC algorithm. We were able to solve instances with up to a size of 9/45 and 203 visits during the week to optimality within 1 h of runtime. For bigger instances with a size of up to 12/60, and 255 visits during the week, good feasible solutions and LBs have been obtained within the time limit of 1 h. Some of the ARC planning units are of this size. This solution approach can, therefore, be used in future to assist the decision makers in these units. For even bigger problem sizes, the BPC algorithm can be used to obtain good LBs in short computing times and may serve as benchmark for (meta)heuristic solution approaches. In our future work, we plan to relax the constraint of visiting clients on prefixed visiting days to obtain more flexibility during the planning. In this case, the visiting days will be implicitly set by the algorithm.

Acknowledgments

The authors thank the Austrian Red Cross, especially Monika Wild and Reinhard Schmidt, for the good cooperation and useful comments on this work. Additionally, the authors are grateful to Ted Ralphs who helped us in understanding the Coin-OR framework and our colleague Klaus-Dieter Rest for proof-reading and useful comments.

REFERENCES

[1] C. Akjiratikarl, P. Yenradee, and P.R. Drake, PSO-based algorithm for home care worker scheduling in the UK, Comput Ind Eng 53 (2007), 559–583.

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

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

[4] A. Bettinelli, A. Ceselli, and G. Righini, A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows, Transp Res Part C 19 (2011), 723–740.

[5] O. Bräysy, W. Dullaert, and N. Pentti, "Municipal routing problems: A challenge for researchers and policy makers?" Bijdragen Vervoerslogistieke Werkdagen 2007, Nautilus Academic Books, Zelzate, Belgium, 2007, pp. 330–347.

[6] D. Bredström and M. Rönnqvist, A branch and price algorithm for the combined vehicle routing and scheduling problem with synchronization constraints, Technical Report 7, NHH Department of Finance & Management Science, Norwegian School of Economics, Norway, 2007.

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

[8] M. Cattafi, R. Herrero, M. Gavanelli, M. Nonato, and F. Malucelli, Improving quality and efficiency in home health care: An application of constraint logic programming for the Ferrara NHS unit, Technical Communications of the 28th International Conference on Logic Programming (ICLP'12), A Dovier and VS Costa, (Editors), Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 17, 2012, pp. 415–424.

[9] A. Ceselli, G. Righini, and M. Salani, A column generation algorithm for a rich vehicle-routing problem, Transp Sci 43 (2009), 56–69.

[10] A. Chabrier, Vehicle routing problem with elementary shortest path based column generation, Comput Oper Res 33 (2006), 2972–2990.

[11] E. Cheng and J.L. Rich, A home health care routing and scheduling problem, Technical Report TR98-04, Department of CAAM, Rice University, Houston Texas, 1998.

[12] G.B. Dantzig and P. Wolfe, Decomposition principle for linear programs, Oper Res 8 (1960), 101–111.

[13] A. Dohn, M.S. Rasmussen, T. Justesen, and J. Larsen, The home care crew scheduling problem, Proc 1st Int Conf Appl Oper Res (ICAOR'08), K. Sheibani (Editor), Yerevan, Armenia, Lecture Notes in Management Science, Vol. 1, 2008, pp. 1–8.

[14] M. Dror, Note on the complexity of the shortest path models for column generation in VRPTW, Oper Res 42 (1994), 977–978.

[15] P. Eveborn, P. Flisberg, and M. Rönnqvist, Laps Care—An operational system for staff planning of home care, Eur J Oper Res 171 (2006), 962–976.

[16] P. Eveborn, M. Rönnqvist, H. Einarsdóttir, M. Eklund, K. Lidén, and M. Almroth, Operations research improves quality and efficiency in home care, Interfaces 39 (2009), 18–34.

[17] D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen, An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems, Networks 44 (2004), 216–229.

[18] M. Gamst and T.S. Jensen, A branch-and-price algorithm for the long-term home care scheduling problem, Oper Res Proc, D. Klatte, H.J. Lüthi, and K. Schmedders (Editors), Springer, 2012, pp. 483–488.

[19] P. Hansen and N. Mladenović, Variable neighborhood search: Principles and applications, Eur J Oper Res 130 (2001), 449–467.

[20] S. Irnich and G. Desaulniers, "Shortest path problems with resource constraints," Column Generation, G. Desaulniers and J. Desrosiers, and M.M. Solomon (Editors), Springer, US, 2005, pp. 33–65.

[21] T. Justesen and M.S. Rasmussen, The home care crew scheduling problem, Master's thesis, Technical University of Denmark and University of Copenhagen, Copenhagen, 2008.

[22] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983), 671–680.

[23] N. Kohl, J. Desrosiers, O.B.G. Madsen, M.M. Solomon, and F. Soumis, 2Path cuts for the vehicle routing problem with time windows, Transp Sci 33 (1999), 101–116.

[24] R. Lougee-Heimer, The common optimization interface for operations research, IBM J Res Dev 47 (2003), 57–66.

[25] S. Nickel, M. Schröder, and J. Steeg, Mid-term and short-term planning support for home health care services, Eur J Oper Res 219 (2012), 574–587.

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

[27] A. Rendl, M. Prandstetter, G. Hiermann, J. Puchinger, and G.R. Raidl, Hybrid heuristics for multimodal homecare scheduling, CPAIOR, Nantes, France, 2012, pp. 339–355.

[28] K.D. Rest and P. Hirsch, A tabu search approach for daily scheduling of home health care services using multi-modal transport, Proc 5th Int Workshop Freight Transp Logist, Odysseus, 2012, pp. 373–376.

[29] K.D. Rest, A. Trautsamwieser, and P. Hirsch, Trends and risk in home health care, J Humanitarian Logist Supply Chain Manage 2 (2012), 34–53.

[30] G. Righini and M. Salani, Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints, Discrete Optim 3 (2006), 255–273.

[31] M. Savelsbergh, The vehicle routing problem with time windows: Minimizing route duration, INFORMS J Comput 4 (1992), 146–154.

[32] J.G. Siek, L.Q. Lee, and A. Lumsdaine, The Boost Graph Library: User Guide and Reference Manual, Addison-Wesley Professional, Indianapolis, Indiana, 2002.

[33] A. Trautsamwieser, M. Gronalt, and P. Hirsch, Securing home health care in times of natural disasters, OR Spectrum 33 (2011), 787–813.