Production, Manufacturing and Logistics

A bi-objective home care scheduling problem: Analyzing the trade-off between costs and client inconvenience

CrossMark

Kris Braekers [a,b,*], Richard F. Hartl [c], Sophie N. Parragh [c], Fabien Tricoire [c]

[a] Research Group Logistics, Hasselt University, Campus Diepenbeek, Agoralaan building D, Diepenbeek 3590, Belgium
[b] Research Foundation Flanders (FWO), Egmontstraat 5, Brussels 1000, Belgium
[c] Department of Business Administration, University of Vienna, Oskar-Morgenstern-Platz 1, Vienna 1090, Austria

ABSTRACT

Organizations providing home care services are inclined to optimize their activities in order to meet the constantly increasing demand for home care. In this context, home care providers are confronted with multiple, often conflicting, objectives such as minimizing their operating costs while maximizing the service level offered to their clients by taking into account their preferences. This paper is the first to shed some light on the trade-off relationship between these two objectives by modeling the home care routing and scheduling problem as a bi-objective problem. The proposed model accounts for qualifications, working regulations and overtime costs of the nurses, travel costs depending on the mode of transportation, hard time windows, and client preferences on visit times and nurses. A distinguishing characteristic of the problem is that the scheduling problem for a single route is a bi-objective problem in itself, thereby complicating the problem considerably. A metaheuristic algorithm, embedding a large neighborhood search heuristic in a multi-directional local search framework, is proposed to solve the problem. Computational experiments on a set of benchmark instances based on real-life data are presented. A comparison with exact solutions on small instances shows that the algorithm performs well. An analysis of the results reveals that service providers face a considerable trade-off between costs and client convenience. However, starting from a minimum cost solution, the average service level offered to the clients may already be improved drastically with limited additional costs.

1. Introduction

In the European Union, the population share of persons older than 60 was 17 percent in 1980 and increased to 22 percent in 2004/5 (it is expected to reach 32 percent in 2030). Life expectancy of men (women) has risen from 68 (76) years to 74 (80) years during the same time period (European Commission, 2007). Increased life expectancy goes hand in hand with increased demand for care. In addition, many elderly people prefer to grow old in the privacy of their homes rather than in a nursing home. On the other hand, willingness for informal care by relatives is decreasing. This is partly due to the fact that women and men are both working (Tarricone & Tsouros, 2008). Therefore, organizations providing home care services are inclined to optimize their activities in order to meet the constantly increasing demand for home care (Koeleman, Bhulai, & van Meersbergen, 2012). This situation gave rise to a number of

publications dealing with the daily routing and scheduling of home care workers in the last couple of years (an overview is given in Table 1). Most of them are application inspired and therefore deal with different variants of the problem.

However, several common characteristics can be identified. First of all, most works consider the total distance traveled or the routing costs of the nurses in the objective function (see e.g. Akjiratikarl, Yenradee, & Drake, 2007; Begur, Miller, & Weaver, 1997; Eveborn, Flisberg, & Rönnqvist, 2006; Eveborn, Rönnqvist, Einarsdóttir, Eklund, Líden, & Almroth, 2009; Hiermann, Prandtstetter, Rendl, Puchinger, & Raidl, 2015; Mankowska, Meisel, & Bierwirth, 2014; Rasmussen, Justesen, Dohn, & Larsen, 2012; Trautsamwieser, Gronalt, & Hirsch, 2011), often in addition to a number of other terms. Besides overtime costs, which can easily be combined with routing costs, these other terms usually account for nurse or client inconvenience. The former aspect involves, e.g. the penalization of assignments to clients that the respective care worker does not like. The latter aspect concerns, e.g. penalties for deviations from preferred visit times or from the set of preferred nurses. Trautsamwieser et al. (2011) consider seven different terms in the objective function and Hiermann et al. (2015) consider as many as 13 (see Table 1, column "# OF terms").

* Corresponding author. Tel.: +32 11269 120; fax: +32 11 268 700.
E-mail addresses: kris.braekers@uhasselt.be (K. Braekers), richard.hartl@univie.ac.at (R.F. Hartl), sophie.parragh@univie.ac.at (S.N. Parragh), fabien.tricoire@univie.ac.at (F. Tricoire).

Table 1
Related work on daily home care worker routing and scheduling problems.

Reference	# OF terms	TW	Skills	Nurse-client	Work time
Akjiratikarl et al. (2007)	1	x	–	–	x
Begur et al. (1997)	1	–	–	–	x
Bertels and Fahle (2006)	5	x	x	x	x
Cheng and Rich (1998)	2	x	x	–	–
Eveborn et al. (2006, 2009)	>7	x	x	x	x
Hiermann et al. (2015)	13	x	x	x	x
Kergosien, Lenté, and Billaut (2009)	1	x	x	x	x
Mankowska et al. (2014)	3	x	x	–	–
Nickel et al. (2012)	4	x	x	x	x
Rasmussen et al. (2012)	3	x	–	x	x
Trautsamwieser et al. (2011)	7	x	x	x	x

A second common characteristic is the consideration of preferences of nurses or clients, e.g. in the objective function as mentioned above. Column "Nurse-Client" in Table 1 indicates whether a paper considers preferences of nurses for clients, preferences of clients for nurses or service consistency, i.e. if an attempt is made to keep the number of different nurses per client low.

Third, a majority of the available studies consider hard time windows on the start of service (see Table 1, column "TW"). In Austria, for example, home care organizations split the day into about five time slots of up to four hours per slot, which usually gives rise to such time windows.

Another common characteristic is the consideration of skills and/or skill levels (Table 1, column "skills"). In most applications, some of the home care workers are registered nurses while others are only qualified for housekeeping or personal hygiene tasks. Finally, in addition to skills, usually also working time regulations are taken into account (see Table 1, column "Work time").

Besides the daily routing and scheduling problem, authors have also addressed the long term problem. Nickel, Schröder, and Steeg (2012) look at weekly schedules and link them to the operational planning problem. Weekly home care scheduling problems are also addressed in, e.g., Borsani, Matta, Beschi, and Sommaruga (2006), Gamst and Jensen (2012), Cappanera and Scutellà (2013), Maya Duque, Castro, Sörensen, and Goos (2015) and Trautsamwieser and Hirsch (2014), while Nowak, Hewitt, and Nataraj (2013) investigate planning horizons of two to three months, anticipating future requests.

Successful implementations of home health care scheduling tools are described, e.g., in Eveborn et al. (2006, 2009) or Begur et al. (1997). An overview of home care routing and scheduling and related problems can be found in Castillo-Salazar, Landa-Silva, and Qu (2015). More information on home care worker scheduling is provided in the survey by Gutiérrez, Gutiérrez, and Vidal (2013) and on personnel scheduling in general by Van den Bergh, Beliën, De Bruecker, Demeulemeester, and De Boeck (2013) and De Bruecker, Van den Bergh, Beliën, and Demeulemeester (2015).

In this paper the focus is on the daily home care routing and scheduling problem. In almost all related studies client inconvenience is either penalized in the objective function or considered in terms of constraints. The first approach assumes that the decision maker is able to provide appropriate weights for each term in the objective function. In the second approach strict bounds on client inconvenience levels have to be respected.

In our opinion, assigning weights to different terms in the objective function a priori can be a difficult task and allowing no deviations from pre-defined client inconvenience levels may be impractical. Thus, the aim of the current paper is to shed some light on the trade-off relationship between cost and client inconvenience in the context of home care routing and scheduling. For this purpose we model the home care routing and scheduling problem as a bi-objective problem. To the best of our knowledge, this is the first time.

The paper is organized as follows. In Section 2, we introduce the bi-objective home care routing and scheduling problem and we devise a mixed integer problem formulation that is strengthened by means of several families of valid inequalities. Embedded into the well-known ϵ-constraint scheme, we solve small instances to optimality. In Section 3, we propose several approaches to solve the scheduling subproblem which is itself a bi-objective problem. We then design a metaheuristic solution framework that is based on multi-directional local search (Tricoire, 2012) to solve instances of realistic size (see Section 4). Finally, we analyze the trade-off between cost and patient inconvenience on a set of new instances in Section 5. These instances are derived from available data from different home health care organizations in Austria and are available online (http://alpha.uhasselt.be/kris.braekers). Conclusions and directions for future research are given at the end of the paper.

2. Problem description

2.1. Problem definition

The Bi-objective Home Care Routing and Scheduling Problem (BI-HCRSP) may be defined as follows. Given a set of nurses and a set of jobs to be performed at patient locations on a single day, the goal is to find a route and schedule for each nurse, indicating the jobs to perform, in which order and at what time.

Nurses have a start and end location (typically their home location), a time window in which they are available to work, and a regular and maximum working time. It is assumed that nurses are paid for their regular working time regardless of the amount of work they do. Working overtime is allowed at a certain cost, although the total working time cannot exceed the maximum. Each nurse uses a certain mode of transportation (e.g. car, public, …), while other nurses may use another mode. Besides, nurses have a certain level of qualification indicating their ability to perform a certain type of job, thereby making some nurse-job combinations infeasible. Lunch breaks for nurses are not considered explicitly, as nurses generally take breaks at their own convenience whenever possible. The time at which a job may be started is restricted by a hard time window. When nurses arrive before the start of the hard time window, they have to wait.

Two objectives are considered: minimizing total costs and minimizing client inconvenience. The former consists of the sum of travel costs and overtime costs of the nurses, while the latter depends on patient preferences regarding nurses and visit times.

Patients may specify preferences regarding the nurses that perform the jobs. For each job, a nurse is indicated as preferred, moderately preferred or not preferred, resulting in a penalty of respectively 0, 1 or 2 when such a nurse is assigned to the job. In addition, for each job, the corresponding patient may indicate a preferred time for the start of service. These preferences are modeled by constructing two additional soft time windows for each job, a tight one and a loose one. When service starts within the tight soft time window, no penalty is

incurred. When service starts outside of the tight soft time window but within the loose one, a penalty of one is incurred. When service starts outside of the loose time window a penalty of two is incurred. Unless stated otherwise, numerical experiments in this paper consider soft time windows with a width of one and two hours, symmetrically around the preferred visit time. However, these values can easily be adapted to reflect other situations as indicated in Section 5.4.

Waiting is only allowed when arriving before the start of a hard time window. Hence, it is not allowed to postpone the start of service of a job to improve the level of patient convenience. The reason is that in reality nurses tend not to wait in such a situation. Modeling time preferences and soft time windows discretely instead of continuously makes the scheduling subproblem of the BIHCRSP slightly easier, by reducing the number of (non-dominated) schedules for a certain route/solution (see Section 3). Furthermore, this way time preferences are expressed similarly as nurse preferences (a penalty of 0, 1 or 2 per job), which makes it more meaningful to aggregate them in a single objective indicating the inconvenience for the patients. Of course, penalty levels can easily be adapted to put more emphasis on one of the objective components. Alternatively, patient convenience regarding time and nurse preference may be considered as two separate objectives. However, in our opinion these are two aspects of the same objective (patient convenience), and hence the decision maker would probably aggregate them anyway. Finally, besides modeling patient preferences, the convenience objective may also be used to incorporate service consistency in this single day problem for jobs which reoccur over a longer planning horizon.

2.2. Problem formulation

We model the BIHCRSP on a directed graph $G = (V, A)$ where V is the set of vertices and A the set of arcs. We consider a set of home care workers $\mathcal{N} = \{1, \ldots, N\}$ and a set of jobs $\mathcal{J} = \{1, \ldots, I\}$. Each job is represented by a separate vertex in our graph, irrespective of whether two or more jobs are associated with the same physical location or client. We denote by 0 the starting location of a nurse and by $I + 1$ the ending depot/location. These may be the same or different physical locations. Thus, $V = \mathcal{J} \cup \{0, I + 1\}$. Parameter q_{in} is used to indicate whether a nurse $n \in \mathcal{N}$ is sufficiently qualified to perform job $i \in \mathcal{J}$ ($q_{in} = 1$) or not ($q_{in} = 0$). Note that $q_{0n} = q_{I+1,n} = 1$. Using this information the arc set is defined as follows: $A = \{(i, j, n) | i \in V \setminus \{I + 1\}, j \in V \setminus \{0\}, n \in \mathcal{N}, i \neq j, q_{in} = 1, q_{jn} = 1\}$.

For each nurse $n \in \mathcal{N}$ a maximum regular working time duration r_n is known, in addition to a maximum allowed daily working time m_n, with $r_n \leq m_n$, and a hard availability time window $[a_n, b_n]$. Working times exceeding r_n incur a cost of d_n per time unit. Finally, let c_{ij}^n and t_{ij}^n denote the travel cost and travel time for nurse n between vertices i and j respectively. These travel costs and travel times are nurse-specific since nurses may use different modes of transportation and their depots may be at different physical locations.

Each job $i \in \mathcal{J}$ has a service duration s_i, a hard time window $[e_i, l_i]$ and a preferred starting time pt_i, where $e_i \leq pt_i \leq l_i$. As a result, the arc set A may be reduced by eliminating arcs between jobs $i, j \in \mathcal{J}$ which are infeasible with respect to hard time windows, i.e. arcs (i, j, n) for which $\max(e_i, a_n + t_{0i}^n) + s_i + t_{ij}^n > \min(l_j, b_n - t_{j,\mathcal{J}+1}^n - s_j)$.

In addition, clients specify preferences for nurses for each job, with pn_{in} indicating the penalty incurred (0, 1 or 2) when assigning nurse n to job i.

In order to formulate the BIHCRSP, we use the following binary decision variables:

$$x_{ij}^n = \begin{cases} 1, & \text{if nurse } n \text{ travels from } i \text{ to } j, \\ 0, & \text{otherwise,} \end{cases}$$

$$p_i^1 = \begin{cases} 1, & \text{if a deviation of more than 30 minutes from } pt_i \text{ exists} \\ & \text{at job } i, \\ 0, & \text{otherwise,} \end{cases}$$

$$p_i^2 = \begin{cases} 1, & \text{if a deviation of more than 60 minutes from } pt_i \text{ exists} \\ & \text{at job } i, \\ 0, & \text{otherwise,} \end{cases}$$

$$z_i = \begin{cases} 1, & \text{if waiting until beginning of time window } e_i \text{ at } i \\ & \text{is necessary,} \\ 0, & \text{otherwise (no waiting is necessary to start job } i), \end{cases}$$

where the last set of variables is used to prohibit waiting inside of hard time windows. We also use the following continuous variables:

$T_0^n = $ time at which nurse n leaves from 0,

$T_{I+1}^n = $ time at which nurse n arrives at $I + 1$,

$T_i = $ time at which service starts at i,

$p_i = $ inconvenience score for job $i \in \mathcal{J}$
    with respect to time windows,

$o_n = $ overtime performed by nurse n.

$$f_1 = \min \sum_{(i,j,n) \in A} c_{ij}^n x_{ij}^n + \sum_{n \in \mathcal{N}} d_n o_n \tag{1}$$

$$f_2 = \min \sum_{(i,j,n) \in A} pn_{in} x_{ij}^n + \sum_{i \in \mathcal{J}} p_i \tag{2}$$

subject to:

$$\sum_{j|(0,j,n) \in A} x_{0j}^n \leq 1 \quad \forall n \in \mathcal{N}, \tag{3}$$

$$\sum_{i|(i,I+1,n) \in A} x_{i,I+1}^n \leq 1 \quad \forall n \in \mathcal{N}, \tag{4}$$

$$\sum_{n,j|(i,j,n) \in A} x_{ij}^n = 1 \quad \forall i \in \mathcal{J}, \tag{5}$$

$$\sum_{j|(j,i,n) \in A} x_{ji}^n = \sum_{j|(i,j,n) \in A} x_{ij}^n \quad \forall i \in \mathcal{J}, n \in \mathcal{N}, \tag{6}$$

$$T_0^n + t_{0j}^n \leq T_j + M_{0j}^n (1 - x_{0j}^n) \quad \forall (0, j, n) \in A, j \neq I + 1,$$
$$M_{0j}^n = b_n + t_{0j}^n - e_j, \tag{7}$$

$$T_i + s_i + \sum_{n|(i,j,n) \in A} t_{ij}^n x_{ij}^n \leq T_j + M_{ij}^1 \left(1 - \sum_{n|(i,j,n) \in A} x_{ij}^n\right) \quad \forall i, j \in \mathcal{J},$$
$$M_{ij}^1 = l_i + s_i - e_j, \tag{8}$$

$$T_i + s_i + t_{i,I+1}^n \leq T_{I+1}^n + M_{i,I+1}^n (1 - x_{i,I+1}^n) \quad \forall (i, I+1, n) \in A, i \neq 0,$$
$$M_{i,I+1}^n = l_i + s_i + t_{i,I+1}^n - a_n, \tag{9}$$

$$e_i \leq T_i \leq l_i \quad \forall i \in \mathcal{J}, \tag{10}$$

$$T_j \leq T_i + s_i + \sum_{n|(i,j,n) \in A} t_{ij}^n x_{ij}^n + M_{ij}^2 \left(1 - \sum_{n|(i,j,n) \in A} x_{ij}^n + z_j\right) \quad \forall i, j \in \mathcal{J},$$
$$M_{ij}^2 = l_j - e_i - s_i, \tag{11}$$

$$T_j \leq e_j + M_j (1 - z_j) \quad \forall j \in \mathcal{J}, \quad M_j = l_j - e_j, \tag{12}$$

$$T_j \leq T_0^n + t_{0j}^n + M_j^n (1 - x_{0j}^n) \quad \forall (0, j, n) \in A,$$
$$M_j^n = l_j - a_n - t_{0j}^n, \tag{13}$$

$$T_{I+1}^n \le T_i + s_i + t_{i,I+1}^n + M_i^n(1 - x_{i,I+1}^n) \quad \forall(i, I+1, n) \in A,$$
$$M_i^n = b_n - e_i - s_i - t_{i,I+1}^n, \tag{14}$$

$$a_n \le T_0^n \le b_n \quad \forall n \in \mathcal{N}, \tag{15}$$

$$a_n \le T_{I+1}^n \le b_n \quad \forall n \in \mathcal{N}, \tag{16}$$

$$T_{I+1}^n - T_0^n \le m_n \quad \forall n \in \mathcal{N}, \tag{17}$$

$$o_n \ge \max(0, T_{I+1}^n - T_0^n - r_n) \quad \forall n \in \mathcal{N}, \tag{18}$$

$$T_i - pt_i \le 30 + p_i^1 M_i^1 \quad \forall i \in \mathcal{J}, \quad M_i^1 = l_i - pt_i - 30, \tag{19}$$

$$T_i - pt_i \ge -30 - p_i^1 M_i^2 \quad \forall i \in \mathcal{J}, \quad M_i^2 = pt_i - e_i - 30, \tag{20}$$

$$T_i - pt_i \le 60 + p_i^2 M_i^3 \quad \forall i \in \mathcal{J}, \quad M_i^3 = l_i - pt_i - 60, \tag{21}$$

$$T_i - pt_i \ge -60 - p_i^2 M_i^4 \quad \forall i \in \mathcal{J}, \quad M_i^4 = pt_i - e_i - 60, \tag{22}$$

$$p_i = p_i^1 + p_i^2 \quad \forall \in \mathcal{J} \tag{23}$$

$$x_{ij}^n \in \{0, 1\} \quad \forall(i, j, n) \in A \tag{24}$$

$$z_i \in \{0, 1\} \quad \forall i \in \mathcal{J} \tag{25}$$

$$p_i^1, p_i^2 \in \{0, 1\}. \quad \forall i \in \mathcal{J} \tag{26}$$

Objective function (1) minimizes the total cost which is composed of routing and overtime costs. Objective function (2) minimizes client inconvenience which is measured by the deviation from the preferred visit time and how disliked the assigned nurses are. Constraints (3) and (4) make sure that each nurse leaves the depot and returns to the depot at most once. Equalities (5) ensure that each job is carried out by a nurse and (6) that each job location is entered and left. Constraints (7)–(10) make sure that the time variables are correctly set and that each job is started within its time window. Waiting times within the time window are prohibited by constraints (11)–(14). Nurses are only allowed to work within a given time window, which is taken care of by constraints (15) and (16). Constraints (17) make sure that the maximum working time is not exceeded and constraints (18) compute the overtime. Client inconvenience in terms of the deviation from the preferred starting time is computed by means of constraints (19)–(23). Finally, constraints (24)–(26) define the domains of the variables.

2.3. Enhancements

To improve the performance of the model described in the previous section, several enhancements are introduced.

Binary assignment variables $y_{in} = \sum_{j \in \mathcal{V}} x_{ij}^n$ may be introduced to indicate whether nurse n is assigned to job i or not. Since each job should be assigned to exactly a single nurse, constraints (27) may then be appended to the model (**Y**). Additionally, branching priority may be given to these assignment variables (**Y$_{BP}$**)

$$\sum_{n \in \mathcal{N}} y_{in} = 1 \quad \forall i \in \mathcal{J}. \tag{27}$$

Furthermore, six families of valid inequalities are considered to further strengthen the model (**IN1-IN6**). Inequalities (28) indicate the relation between variables p_i^1 and p_i^2 (**IN1**) while inequalities (29) and (30) exclude subtours of length two and three respectively (**IN2-IN3**)

$$p_i^2 \le p_i^1 \quad \forall i \in \mathcal{J} \tag{28}$$

$$\sum_{n \in \mathcal{N}} (x_{ij}^n + x_{ji}^n) \le 1 \quad \forall i, j \in V, n \in \mathcal{N} \tag{29}$$

$$\sum_{n \in \mathcal{N}} (x_{ij}^n + x_{ji}^n + x_{ik}^n + x_{ki}^n + x_{jk}^n + x_{kj}^n) \le 2 \quad \forall i, j, k \in V, n \in \mathcal{N}. \tag{30}$$

Partial routes $< i - j - k >$ between three vertices $i, j, k \in \mathcal{J}$ which are infeasible for a nurse n due to time windows may be excluded as well (**IN4**). Combined with the fact that subtours are not allowed, this yields inequality (31). When all partial routes between vertices $i, j, k \in \mathcal{J}$ are infeasible for nurse n, the inequality may be strengthened as shown in (32). When one of these inequalities is valid for several nurses, it may be strengthened by including the respective arcs of each nurse on the left-hand side of (31) or (32) respectively

$$x_{ij}^n + x_{ji}^n + x_{jk}^n \le 1 \quad \forall i, j, k \in \mathcal{J}, n \in \mathcal{N} | e_i + s_i + t_{ij}^n + s_j + t_{jk}^n > l_k \tag{31}$$

$$x_{ij}^n + x_{ji}^n + x_{jk}^n + x_{kj}^n + x_{ik}^n + x_{ki}^n \le 1 \quad \forall i, j, k \in \mathcal{J},$$
$$n \in \mathcal{N} | \text{ all partial routes infeasible.} \tag{32}$$

The relation between routing variables x_{ij}^n and variables p_i and p_j indicating the inconvenience related to the timing of service is expressed in inequalities (33)–(36) (**IN5**). Again these may be strengthened when valid for multiple nurses

$$2x_{ij}^n \le p_i + p_j \quad \forall(i, j, n) \in A | pt_i - 30 + s_i + t_{ij}^n > pt_j + 60 \tag{33}$$

$$x_{ij}^n \le p_i + p_j \quad \forall(i, j, n) \in A | pt_i - 30 + s_i + t_{ij}^n > pt_j + 30 \tag{34}$$

$$2x_{ij}^n \le p_i + p_j \quad \forall(i, j, n) \in A | pt_i + 30 + s_i + t_{ij}^n$$
$$\le pt_j - 60 \wedge e_j < pt_j - 60 \tag{35}$$

$$x_{ij}^n \le p_i + p_j \quad \forall(i, j, n) \in A | pt_i + 30 + s_i + t_{ij}^n$$
$$\le pt_j - 30 \wedge e_j < pt_j - 30. \tag{36}$$

Finally, variables x_{ij}^n and p_i (or p_j) may be combined with the time windows to yield inequalities (37)–(42) (**IN6**)

$$2x_{ij}^n \le p_i \quad \forall(i, j, n) \in A | pt_i - 60 + s_i + t_{ij}^n > l_j \tag{37}$$

$$x_{ij}^n \le p_i \quad \forall(i, j, n) \in A | pt_i - 30 + s_i + t_{ij}^n > l_j \tag{38}$$

$$2x_{ij}^n \le p_j \quad \forall(i, j, n) \in A | e_i + s_i + t_{ij}^n > pt_j + 60 \tag{39}$$

$$x_{ij}^n \le p_j \quad \forall(i, j, n) \in A | e_i + s_i + t_{ij}^n > pt_j + 30 \tag{40}$$

$$2x_{ij}^n \le p_j \quad \forall(i, j, n) \in A | e_j < l_i + s_i + t_{ij}^n < pt_j - 60 \tag{41}$$

$$x_{ij}^n \le p_j \quad \forall(i, j, n) \in A | e_j < l_i + s_i + t_{ij}^n < pt_j - 30. \tag{42}$$

Computational experiments in Section 5.2 indicate the effect of using the enhancements described above. Note that the valid inequalities are added before the model is solved whenever a violation is possible (taking into account the reduced arc set). We did not consider adding the inequalities in a branch-and-cut fashion as it is not our intention to provide a state-of-the-art exact method to solve the problem at hand. The mathematical model is used to acquire some initial insights into the problem and as a tool to assess the qual-

ity of the heuristic method proposed in Section 4 on small problem instances.

We note here that, while the model does not require any assumptions on the input data, we assume that all time related parameters are integer and we exploit this property in the scheduling algorithm described next.

3. Scheduling problem

This section focuses on the scheduling subproblem of the BI-HCRSP. Let an unscheduled solution denote a set of unscheduled routes (or job sequences), one for each nurse, indicating the patients to be visited by this nurse and their visiting order. The scheduling problem then consists of deciding on a time schedule for each of these routes such that all constraints are satisfied.

Due to the bi-objective nature of our problem and the numerous side constraints, the scheduling problem is non-trivial. A distinctive characteristic of our scheduling problem is the fact that the scheduling of a single route is already a bi-objective problem in itself as explained below. Hence, a single unscheduled route may represent multiple non-dominated scheduled routes, and an unscheduled solution may represent many mutually non-dominated solutions to the BIHCRSP. Therefore the scheduling problem may be decomposed into two problems which can be solved sequentially: generating all non-dominated schedules for each individual route (Section 3.1) and generating all non-dominated solutions from the set of schedules for each route (Section 3.2).

3.1. Scheduling of a single route

The scheduling of a single route (or sequence of jobs) involves deciding on the visit time of each job. Since waiting is only allowed before hard time windows, this can be reduced to determining the start time of the route (the time the nurse leaves their starting location). For a given start time, all other timing variables of the route can easily be calculated. As mentioned above, a distinctive characteristic of this problem is the fact that it is bi-objective in itself. The route schedule may influence both the amount of overtime (and hence total costs) and how well time preferences are satisfied (and hence the level of inconvenience). While postponing the start of the route as much as possible will guarantee the minimization of overtime costs, the effect on the level of inconvenience might be positive or negative. In fact, the relationship between the start time of a route and the level of inconvenience will often be nonlinear (e.g. with increasing start time of the route, the inconvenience level may first increase and then decrease). As a result, solving the scheduling problem involves finding the set of non-dominated schedules and their corresponding start time of the route. This is in contrast with other scheduling problems as subproblems of routing problems in the literature, which are generally either constraint-satisfaction problems (e.g. finding a schedule which satisfies hard time windows) or single objective problems (e.g. minimizing soft time window violations or minimizing route duration).

Using the notation of Vidal, Crainic, Gendreau, and Prins (2015), the scheduling problem may be described as in (43)

$$\{DUR \cup \sum_i c_i(t_i) | TW, DUR, P(t)\}. \tag{43}$$

The objectives consist of a route duration feature to minimize overtime costs (DUR) and a sum of non-convex time-dependent cost functions to minimize the level of inconvenience ($\sum_i c_i(t_i)$). Since the latter functions are piecewise linear, they can be optimized efficiently (despite being non-convex) using dynamic programming (Vidal et al., 2015). Constraints include hard time windows (TW) and a duration constraint to satisfy maximum working times (DUR), while the no-waiting constraints may be considered as time-dependent processing times ($P(t)$).

As the scheduling problem is likely to be solved a considerable number of times in a heuristic solution approach, the problem should be solved efficiently. Several approaches have been considered by the authors. Preliminary tests indicated that using CPLEX to solve the scheduling problem is too time consuming. This may be explained by the fact that a mathematical formulation of the scheduling problem still contains binary decision variables due to the combination of the hard time window and the no-waiting constraints.

Since only integer values are considered for all time-related parameters such as travel times, time windows, preferred visit times, etc. (see Section 5.1), a simple enumeration method has been tested. This method consists of performing a forward loop through the route for each feasible start time of the route to determine the corresponding overtime costs and inconvenience level. At the end, dominated schedules can easily be removed. An improved version of the enumeration method is considered as well. In that case, the search starts with the latest feasible start time of the route. During each forward loop through the route, the minimal time by which the start time of the route should be decreased in order to improve (i.e. reduce) the inconvenience level at one of the jobs is maintained. In the next iteration, the start time of the route is decreased by this value since all feasible start times in between may be discarded (the inconvenience level will not improve and decreasing the start time of the route may never have a positive effect on overtime costs).

Finally, a dynamic programming method is proposed. The idea behind this method is partially based on existing methods for non-convex piecewise linear cost functions (for an overview we refer to Vidal et al. (2015) and Hashimoto, Yagiura, Imahori, and Ibaraki (2013)). However, these methods only deal with a single objective (minimizing the total cost function) and therefore cannot be applied directly in our bi-objective setting. Given a route which is feasible with respect to time windows, our method consists of a single forward loop through the route, while maintaining for each node a list of time intervals in which the penalty with respect to time preferences stays the same. More specifically, for each node a number of "transition points" are calculated, corresponding to a service start time just before the total penalty for the partial route up to this node changes, i.e. starting service 1 minute later at this node will result in a change in the total penalty up to this node.

The pseudo code for this method is presented in Algorithm 1. The following notation is used. A route $< v_0, v_1, v_2, \ldots, v_k, v_{k+1} >$ is considered with k jobs and v_0 and v_{k+1} denoting the depot location of the nurse n that is assigned to this route (node indices i, j). Let E_i and L_i represent the earliest and latest time that service at node v_i may start to guarantee time window feasibility for the complete route (see also Section 4.3.2). A set of transition points for node v_i is represented by $\mathcal{T}^i = \{T_1^i, T_2^i, \ldots, T_{|\mathcal{T}^i|}^i\}$. For each transition point $T_f^i \in \mathcal{T}^i$, a tuple of three values (t_f^i, p_f^i, w_f^i) is stored, indicating respectively the start of service at node v_i, the total penalty up to node v_i when starting service at time t_f^i and the latest possible start time at the depot that corresponds to the previous two values. Similarly, let $\mathcal{T}^j$ and $\mathcal{T}^{\text{temp}}$ denote respectively the set of transition points for node v_j (index h) and a temporary set of transition points (index g). A transition point $T_h^j \in \mathcal{T}^j$ is defined by tuple (t_h^j, p_h^j, w_h^j), while a transition point $T_g^{\text{temp}} \in \mathcal{T}^{\text{temp}}$ is defined by tuple $(t_g^{\text{temp}}, p_g^{\text{temp}}, w_g^{\text{temp}})$. Furthermore, let $\mathcal{T}^0 = \{(b_n, 0, b_n)\}$ and $\rho^j(t)$ a function to calculate the penalty regarding time preferences when service starts at time t at node v_j.

In the first part of Algorithm 1 (lines 1–21), for each job v_j ($j = 1, \ldots, k$) the set of transition points $\mathcal{T}^j$ is calculated, starting from the set of transition points $\mathcal{T}^i$ of the previous node v_i in the route and a set of temporary transition points $\mathcal{T}^{\text{temp}}$. First, the latter set $\mathcal{T}^{\text{temp}}$ for job v_j is generated as follows (line 5). Starting from an empty set, a transition point T_g^{temp} is added for each point in time right before the penalty with respect to time preferences would change ($pt_{v_j} - 61$,

Algorithm 1 Scheduling method.

```
 1: //find the set of transition points for each job in the route
 2: 𝒯⁰ = {(bₙ, 0, bₙ)}
 3: for j = 1 → k do
 4:    i = j − 1
 5:    generate 𝒯ᵗᵉᵐᵖ
 6:    𝒯ʲ = ∅, f = g = 1, h = 0
 7:    while f ≤ |𝒯ⁱ| and g ≤ |𝒯ᵗᵉᵐᵖ| do
 8:       h = h + 1
 9:       if tᶠⁱ + sᵥᵢ + tⁿᵥᵢᵥⱼ ≤ tᵍᵗᵉᵐᵖ then
10:          tₕʲ = max(tᶠⁱ + sᵥᵢ + tⁿᵥᵢᵥⱼ, eᵥⱼ)
11:          𝒯ʲ = 𝒯ʲ ∪ {Tₕʲ(tₕʲ, pᶠⁱ + ρʲ(tₕʲ), wᶠⁱ)}
12:          if tᶠⁱ + sᵥᵢ + tⁿᵥᵢᵥⱼ = tᵍᵗᵉᵐᵖ then
13:             g = g + 1
14:          end if
15:          f = f + 1
16:       else
17:          𝒯ʲ = 𝒯ʲ ∪ {Tₕʲ(tᵍᵗᵉᵐᵖ, pᶠⁱ + ρʲ(tᵍᵗᵉᵐᵖ), wᶠⁱ − (tᶠⁱ + sᵥᵢ +
             tⁿᵥᵢᵥⱼ − tᵍᵗᵉᵐᵖ))}
18:          g = g + 1
19:       end if
20:    end while
21: end for
22: //find the transition points for the end depot vₖ₊₁
23: 𝒯ᵏ⁺¹ = 𝒯ᵏ
24: for f = 1 → |𝒯ᵏ⁺¹| do
25:    Tᶠᵏ⁺¹(tᶠᵏ⁺¹, pᶠᵏ⁺¹, wᶠᵏ⁺¹) = Tᶠᵏ⁺¹(tᶠᵏ⁺¹ + sᵥₖ + tⁿᵥₖᵥₖ₊₁, pᶠᵏ⁺¹, wᶠᵏ⁺¹)
26: end for
27: //find the set of non-dominated schedules
28: 𝒮 = ∅, s = 1, f = |𝒯ᵏ⁺¹|
29: if tᶠᵏ⁺¹ − wᶠᵏ⁺¹ ≤ mₙ then
30:    oₛ = max(0, tᶠᵏ⁺¹ − wᶠᵏ⁺¹ − rₙ)
31:    pₛ = pᶠᵏ⁺¹
32:    𝒮 = 𝒮 ∪ {Sₛ(oₛ, pₛ)}
33:    f = f − 1
34:    while f > 0 do
35:       if pᶠᵏ⁺¹ < pₛ then
36:          s = s + 1
37:          oₛ = max(0, tᶠᵏ⁺¹ − wᶠᵏ⁺¹ − rₙ)
38:          if oₛ ≤ mₙ − rₙ then
39:             if oₛ = oₛ₋₁ then
40:                𝒮 = 𝒮 \ {Sₛ₋₁} ∪ {Sₛ(oₛ, pᶠᵏ⁺¹)}
41:             else
42:                𝒮 = 𝒮 ∪ {Sₛ(oₛ, pᶠᵏ⁺¹)}
43:             end if
44:          else
45:             f = 0
46:          end if
47:       end if
48:       f = f − 1
49:    end while
50: end if
```

Table 2
Comparison of scheduling methods.

Method	Computation time (s)
CPLEX	>3600
Enumeration	0.494
Enumeration improved	0.317
Dynamic programming	0.165

its first transition point ($t_1^i + s_{v_i} + t_{v_i v_j}^n < e_{v_j}$). Although in this case the penalty level does not change, the transition point is required to account for the fact that when nurse n arrives at node v_j before time e_{v_j}, the nurse should wait and hence route duration is affected. The transition points $T_g^{\text{temp}} \in \mathcal{T}^{\text{temp}}$ are sorted from small to large according to t_g^{temp} and for each transition point the penalty level is $p_g^{\text{temp}} = \rho^j(t_g^{\text{temp}})$, while the value of w_g^{temp} is irrelevant. Second, iteratively the smallest transition point among $\mathcal{T}^i$ (increased with the time required to reach v_j) and $\mathcal{T}^{\text{temp}}$ is selected and a new transition point for v_j is created (lines 7–21). In case the transition point in $\mathcal{T}^{\text{temp}}$ is selected (lines 16–18), note that the start time of the route in the newly created transition point T_h^j is equal to the start time of the route for the transition point in $\mathcal{T}^i$ minus the difference between the start of service at job v_j for both transition points, i.e. $w_h^j = w_f^i - (t_f^i + s_{v_i} + t_{v_i v_j}^n - t_g^{\text{temp}})$.

When the transition points for the final job in the route $\mathcal{T}^k$ have been found, the transition points for the end depot $\mathcal{T}^{k+1}$ are calculated by copying $\mathcal{T}^k$ and increasing the start of service of each of these transition points with the service time at v_k and the time to travel to the end depot v_{k+1} (lines 22–26). Finally, the set of non-dominated schedules $\mathcal{S} = \{S_1(o_1, p_1), \ldots, S_{|\mathcal{S}|}(o_{|\mathcal{S}|}, p_{|\mathcal{S}|})\}$ (index s) is found, where each schedule S_s is represented by an amount of overtime o_s and a penalty level regarding time preferences p_s (lines 27–50). This is done by considering each of the transition points of v_{k+1} in reverse order, i.e. starting with the final one in $\mathcal{T}^{k+1}$ (the one with the largest arrival time at the end depot and hence also the latest corresponding start time of the route), and calculating the corresponding overtime costs. Note that when considering the transition points in this order, overtime costs cannot decrease since overtime is minimal when starting the route as late as possible. Hence, only transition points which reduce the penalty level should be considered. Furthermore, the search can be stopped whenever a transition point violates the maximum working time constraint.

Table 2 compares computation times of the different scheduling methods. Each method was used to schedule all routes of all non-dominated solutions that were found in five runs of the base algorithm described in Section 4 on all 90 benchmark instances. This corresponds to 14,153 solutions consisting of 722,913 routes in total. Total computation times over all routes are reported. Solving the single route scheduling problem by CPLEX is clearly too time consuming. The other approaches are considerably faster, scheduling all routes in less than half a second. While the proposed improvements of the enumeration method reduce its computation time, the dynamic programming approach clearly performs best. Therefore, this method has been used in all other experiments described in this paper.

3.2. Scheduling of multiple routes

Given a set of non-dominated schedules for each route of an unscheduled solution, a scheduled solution is obtained by selecting a single schedule for each route. To obtain only the non-dominating solutions, a simple dynamic programming-based method is applied. In a first step, the routes are sorted according to their number of non-dominated schedules, from small to large. Second, all combinations of the schedules of routes one and two are made and the dominated

$pt_{v_j} - 31$, $pt_{v_j} + 30$, $pt_{v_j} + 60$), provided that it satisfies the condition $E_j \leq t_g^{\text{temp}} < L_j$. A transition point for start of service at time L_j is added as well, since the penalty value increases to infinity after this time. Finally, an additional transition point is added for time of service equal to e_{v_j} when (1) no such transition point already exists ($\nexists T_g^{\text{temp}} \in \mathcal{T}^{\text{temp}} | t_g^{\text{temp}} = e_{v_j}$), (2) job v_j may be started as early as the start of its time window ($E_j = e_{v_j}$), and (3) waiting time will exist at job v_j when starting service at the preceding node v_i at the time of

ones are discarded. Next, the non-dominated combinations are combined with the schedules from route three and so on, until all routes have been considered.

4. Metaheuristic approach

Due to the bi-objective nature of the BIHCRSP, a single optimal solution to the problem will often not exist. Instead, the goal is to find the set of Pareto optimal or efficient solutions. It is assumed that the reader is familiar with the basic concepts of multi-objective optimization such as Pareto optimality and dominance. For a detailed description of these concepts and their underlying principles, the reader is referred to Ehrgott and Gandibleux (2002, 2004) and Ehrgott (2005).

Since exactly solving instances of realistic size in a reasonable amount of computation time does not seem feasible, a metaheuristic algorithm is proposed to find a set of mutually non-dominated solutions which approximates the set of efficient solutions. The algorithm is based on the multi-directional local search framework (Tricoire, 2012) and uses large neighborhood search (LNS) as a subheuristic.

Multi-directional local search (MDLS) is a recently proposed meta-heuristic framework for multi-objective optimization problems (Tricoire, 2012). It is based on the idea that in order to find new efficient solutions that are neighbors of a solution x, it is sufficient to start a search from x in the direction of one objective at a time only. In general, the method works as follows. An archive of non-dominated solutions is maintained and in each iteration of the algorithm a solution is selected from this archive. For each objective a single-objective local search is performed on the selected solution. The resulting new solutions, as many as there are objectives, are then used to update the archive. Advantages of the method are its simplicity, flexibility and the fact that for each objective any existing single-objective local search method may be applied.

Large neighborhood search (LNS) is a metaheuristic which was first introduced by Shaw (1998). It uses the concept of *ruin and recreate* to define an implicit, large neighborhood of a current solution as the set of solutions that may be attained by destroying a large part of the solution and subsequently rebuilding the resulting partial solution. A successful general-purpose LNS algorithm for a variety of vehicle routing problems was proposed by Pisinger and Ropke (2007). The algorithm iteratively removes a number of customers from the current solution and reinserts them to obtain a new solution. Several simple removal and insertion operators, selected randomly in each iteration, are applied. An adaptive version of LNS is proposed by Ropke and Pisinger (2006) in which the selection of the operators is biased using their success in previous iterations. In recent years, many routing problems have been successfully solved using LNS-based methods. For details and an overview of recent developments on LNS, the reader is referred to Pisinger and Ropke (2010).

Tricoire (2012) shows that, using LNS as a subheuristic, the MDLS framework produces results which are competitive to those of the best known solution method for three general multi-objective optimization problems (multi-dimensional multi-objective knapsack problem, bi-objective set packing problem, bi-objective orienteering problem). Therefore, a similar approach is proposed for the BIHCRSP. The general structures of the MDLS algorithm and LNS subheuristic are discussed in Sections 4.1 and 4.2, while the LNS operators and their implementation are discussed in Section 4.3.

4.1. MDLS structure

The general structure of the MDLS algorithm is presented in Algorithm 2. The initial set of solutions, which serves as an input for the MDLS algorithm, is found by applying each insertion operator (see Section 4.3.2) individually on the problem. In each iteration of the algorithm, a solution x is randomly selected from F. Next, for each

---

Algorithm 2 MDLS structure.

input: a set of non-dominated scheduled solutions F
repeat
  $x \leftarrow select\_a\_solution(F)$
  $G_{cost} \leftarrow$ set of solutions generated by $LNS_{cost}(x)$
  $G_{incon} \leftarrow$ set of solutions generated by $LNS_{incon}(x)$
  $G \leftarrow G_{cost} \cup G_{incon}$
  $update(F, G)$
until stopping criterion is met
return F

---

objective an LNS iteration is performed, resulting in a set of new solutions G. These new solutions are used to update set F. This is repeated until the stopping criterion is met, which can either be a predefined number of iterations, a maximum computation time or any other criterion defined by the decision maker. The structure of our MDLS algorithm is very similar to the structure described by Tricoire (2012), although in our case a single-objective local search procedure may result in more than one new solution as is discussed in Section 4.2. A set of non-dominated scheduled solutions F is maintained throughout the search, i.e. for each solution in F the timing variables are fixed to specific values. These values are required to know the actual objective values of the solution, but as a result several solutions in the set may have the same routing (but a different timing). Set F is stored as an ordered list which reduces the number of dominance checks to be performed when updating the set compared to an unordered list (see Tricoire, 2012). Besides, in order to diversify the search, solutions which have the same objective values but a different routing are all kept.

For hard instances, it may take a number of iterations to find a feasible solution. This is handled by using a request bank, which is a common concept in LNS methods. Whenever not all jobs can be inserted in the routes of the nurses, the remaining jobs are put into the request bank. New solutions are first evaluated on the number of jobs in the request bank and second on both objective values. This means that as soon as a feasible solution to the problem has been found, infeasible solutions are no longer allowed during the search and these may be discarded immediately.

4.2. LNS structure

Two major issues arise in the implementation of any local search move for the problem under study (no matter whether using small or large neighborhoods). First, the objective values of a new solution can only be measured when the scheduling subproblem is solved. As discussed in Section 3, this problem is non-trivial and although an efficient algorithm has been proposed, it seems unpractical to solve the scheduling problem from scratch after each local search move as this would result in excessive computation times. Second, even when developing an efficient reoptimization algorithm by maintaining non-dominated schedules for partial routes to avoid having to solve the scheduling problem from scratch, the evaluation of local search moves (or of insertion positions of a job in LNS) and selecting the best one is not straightforward, since a single new routing solution may result in multiple non-dominated scheduled solutions.

For each objective (costs and inconvenience), a single LNS iteration in the direction of this objective is performed on the selected solution x (Algorithm 3). First, the number of jobs to be removed in this iteration (q), and the removal and insertion operators to be applied (a and b) are determined randomly. Next, using removal operator a, q jobs are removed from x and added to the request bank, resulting in a partial solution x'. The jobs which have been removed (and those potentially already in the request bank) are then reinserted using the selected insertion operator b. To overcome the issues stated above,

Algorithm 3 LNS structure.

input: solution x
$q \leftarrow$ number of jobs to be removed
$a \leftarrow$ randomly selected removal operator
$b \leftarrow$ randomly selected insertion operator
$x' \leftarrow removal(x, q, a)$
$x' \leftarrow insertion(x', b)$
if $rb(x') \leq rb(x)$ **then**
    $\mathcal{S} \leftarrow$ non-dominated schedules of x'
    return $\mathcal{S}$
else
    return $\emptyset$
end if

it is decided not to perform these LNS iterations on scheduled solutions, but only on the routing aspect of a solution. This means that jobs are inserted without solving the scheduling problem and hence without knowing the non-dominated set of actual objective values that result from this insertion. Instead, insertion positions are evaluated based on approximative objective values which are discussed in Section 4.3.2. Only after all jobs have been inserted (or no feasible insertion positions for the remaining jobs exist), the scheduling problem is solved to obtain the set of non-dominated solutions from the newly obtained routing solution using the method proposed in Section 3. Of course, this final step may be skipped and the new routing solution x' may be discarded immediately when the number of jobs in the request bank $rb(x')$ is larger than that of the solutions currently in set F.

Note that performing multiple LNS iterations on a single solution is not considered as preliminary results indicated that the second objective is likely to diverge too far after a few iterations, making it unlikely to find new non-dominated solutions.

4.3. LNS operators

Several standard removal and insertion operators from the LNS literature (Pisinger & Ropke, 2007; Ropke & Pisinger, 2006; Shaw, 1998) are adapted to the specific problem context of the BIHCRSP. They are discussed in Sections 4.3.1 and 4.3.2 respectively.

4.3.1. Removal operators

For each objective, six removal operators are considered. Three of them are the same for both objectives (random, route, related) while the other three represent objective-specific implementations of the worst removal concept.

The *random* removal operator randomly removes q jobs from the current solution, while the *route* removal operator removes complete routes which are randomly selected until q jobs have been removed. The *related* removal operator removes jobs which are related to each other and hence are expected to be easy to interchange. The relatedness r_{ij} of two jobs $i, j \in \mathcal{J}$ is expressed in terms of their location, preferred timing, qualification requirements or a random weighted combination of these (44), each of these options having an equal probability of being chosen (i.e. (α, β, γ) is $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$ or any random combination with $\alpha + \beta + \gamma = 1$ and all three non-negative). Relatedness in terms of location is based on the average travel time over all nurses between the two jobs, while relatedness in terms of qualification requirements is based on the number of nurses which are only qualified to perform one of the jobs. A lower value of r_{ij} indicates that jobs i and j are more related. Note that t'^n_{ij} and pt'_i represent normalized values such that they only take values from [0, 1]. The implementation of the operator is the same as in Shaw (1998). Initially a job is removed randomly. Next, iteratively an already removed job is selected randomly and its most related job

(of those still in the solution) is removed

$$r_{ij} = \alpha \sum_{n \in \mathcal{N}} t'^n_{ij}/N + \beta|pt'_i - pt'_j| + \gamma \sum_{n \in \mathcal{N}} |q_{in} - q_{jn}|/N. \qquad (44)$$

For the objective of minimizing costs, a *worst travel cost*, a *worst minimal overtime cost* and a *worst combined cost* removal operator are applied. The first two operators remove jobs which result in respectively the largest travel cost savings and the largest minimal overtime cost savings when being removed from the solution. The minimal overtime cost of a route is defined as the overtime cost when starting the route as late as possible and hence resulting in the smallest possible amount of overtime. The minimal overtime cost is used instead of the actual overtime cost as the latter would require resolving the scheduling problem when removing a job while the former may be calculated in constant time (Vidal et al., 2015). The third operator removes jobs which result in the largest cost savings when the travel costs and minimal overtime costs are combined.

Note that in our problem context some (or all) nurses may use public transportation and hence their traveling will not result in travel costs (see Section 5.1). Therefore, the term "travel costs" is interpreted slightly different for the worst removal operators. When all nurses use public transportation, the effect on travel time is used instead. When only some nurses use public transportation, the cost of traveling by car is used for all nurses.

For the objective of minimizing inconvenience, similarly a *worst nurse inconvenience*, *worst time inconvenience* and *worst total inconvenience* removal operator are applied. The effect of removing a job on total time inconvenience of a route is not modeled as the difference in time inconvenience between the route with and without the job, as this would again involve solving the scheduling problem to find the second value. Instead, the effect on time inconvenience of removing job i is only based on the penalty value for time inconvenience (p_i) of this job in the current solution.

As in Shaw (1998) and Ropke and Pisinger (2006), a parameter $P \geq 1$ is used in all worst removal operators to introduce some randomness in the selection of jobs, thereby avoiding the same jobs to be removed over and over again. In each iteration of a removal operator, the list of L jobs which may be removed is ordered from worst to best, and the xth job in the list is selected to be removed with $x = \lfloor y^P L \rfloor$ and y a random number in [0, 1]. A lower value of P corresponds to more randomness. Besides, we add a very small random noise value to the savings of removing a job to randomize the order in which jobs with the same savings are selected (as in our problem context often several jobs may result in the same savings, especially with respect to the inconvenience objective).

4.3.2. Insertion operators

Four insertion operators are applied. The structure of these operators is the same for both objectives, although the considered objective function differs.

The well-known *basic greedy*, *regret-2* and *regret-3* operators are applied. The former iteratively inserts the job with the cheapest insertion position among all jobs to be inserted, while the regret-k operators take into account the difference in insertion cost between the least-cost route and the next $k - 1$ least-cost routes. For a detailed discussion of these operators, the reader is referred to Ropke and Pisinger (2006) and Pisinger and Ropke (2007). Furthermore, an other greedy operator, denoted *random greedy*, is applied. This operator iteratively selects a job to be inserted randomly and inserts it at its cheapest insertion position.

Note that the feasibility of an insertion position can be evaluated in constant time. Parameter q_{in} indicates whether nurse n is qualified to carry out job i while time window and route duration constraints may be evaluated in constant time by maintaining for each job $i \in \mathcal{J}$ the earliest time service may start (E_i) and the latest time

service should start (L_i) in order for the rest of the route to be feasible (Campbell & Savelsberg, 2004; Kindervater & Savelsbergh, 1997).

Since we do not want to solve the scheduling problem for each feasible insertion position of each job to be inserted, the actual objective values cannot be used to evaluate insertion positions. Instead approximative objective functions are applied. For this matter we decided to use several hierarchical objectives for two reasons. First, this allows to discriminate between insertion positions with the same primary insertion cost (as is often the case due to the absence of travel costs when using public transportation and the limited penalty levels for the inconvenience objectives). Second, this allows to take the cost objective into account as a tie-breaker when minimizing inconvenience (and vice versa).

For the objective of minimizing costs, three hierarchically structured objectives are considered. The first objective is to minimize the sum of travel costs and minimal overtime costs, approximating the actual objective. The second and third objective are to minimize route duration and total travel time respectively. The idea is that it is beneficial to construct compact routes with small travel and waiting times, even when no travel costs are incurred and insertion positions do not generate overtime costs, as this would allow other jobs to be inserted later at lower costs. Preliminary results have indicated that using the approximate objective value for inconvenience as a fourth objective or a third objective (instead of travel time) has only limited impact. Hence, this is not considered.

For the objective of minimizing inconvenience, two hierarchical objectives are used to evaluate an insertion position of a job, an approximation of the total inconvenience incurred by this job and the sum of the additional travel costs and minimal overtime costs. Total inconvenience $incon_i$ when inserting job i between jobs $i-1$ and $i+1$ in the route of nurse n is approximated as in (45). A penalty value in the interval $[0, 4]$ is obtained. The first term is the inconvenience with respect to the nurse. The second term approximates the inconvenience with respect to the visit time and consists of the average of two measures. The first measure (η) indicates whether for job i a feasible visit time $t \in [E_i, L_i]$ exists within either the tight soft time window (0), the loose soft time window (1), or none of both (2). The second part indicates whether the preferred visit times of nodes $i-1$, i and $i+1$ are in increasing order or not

$$incon_i = pn_{in} + (\eta + \theta)/2$$

with

$$\eta = \begin{cases} 0 & \text{if } E_i \leq pt_i + 30 \text{ and } L_i \geq pt_i - 30 \\ 1 & \text{if } E_i \leq pt_i + 60 \text{ and } L_i \geq pt_i - 60 \text{ and} \\ 2 & \text{else} \end{cases}$$

$$\theta = \begin{cases} 0 & \text{if } pt_{i-1} \leq pt_i \text{ and } pt_i \leq pt_{i+1} \\ 2 & \text{if } pt_{i-1} > pt_i \text{ and } pt_i > pt_{i+1} \\ 1 & \text{else} \end{cases} \quad (45)$$

To diversify the search, a noise term may be added to the objective functions of the insertion heuristics (Pisinger & Ropke, 2007; Ropke & Pisinger, 2006). At each LNS iteration, we select randomly whether to apply noise or not. To account for the fact that hierarchical objectives are used, noise is only added to the sum of travel costs and minimal overtime costs, and to total route duration when the original objective value differs from zero (to avoid losing the effect of the second or third objective). Furthermore, adding noise to $incon_i$ is modeled by defining small probabilities for adding (subtracting) 0.5 or 1 to (from) $incon_i$.

5. Computational results

All algorithms are implemented in C++. MDLS is run on an Intel Xeon Processor E5-2670 at 2.50 gigahertz, using a single thread. To solve the model, ILOG Cplex 12.5 is used. All experiments with the model are performed on the Vienna Scientific Cluster (VSC-1) using

Intel X5550 CPUs at 2.66 gigahertz and a run time limit of two days. In the next sections, we first present the characteristics of our benchmark instances and then the results of our numerical experiments.

5.1. Problem instances

Since no benchmark data are available for our problem, problem instances have been generated randomly. However, the parameter values that have been applied are based on both real-life data of two Viennese companies and real-life-based benchmark data for a related problem (Hiermann et al., 2015). A general discussion of these instances is provided here, while numerical details are available in Appendix A. The instances are available online (http://alpha.uhasselt.be/kris.braekers).

A set of 90 instances has been generated, consisting of a first set of 30 small test instances (10–25 jobs) and a second set of instances of realistic size (50–300 jobs). For each instance, the set of nurses consists of a number of full-time nurses with a regular and maximum working time of 8 and 10 hours respectively, and a number of part-time nurses with a regular and maximum working time of 4 and 6 hours respectively (either in the morning or the evening). The number of nurses is set such that on average the number of jobs is five times the number of full-time nurse equivalents.

Six types of jobs are considered, corresponding to six qualification levels of the nurses. These nurse qualification levels are assumed to be non-hierarchical since high-qualified nurses generally do not perform jobs requiring a low qualification level. Probabilities for a job to require a certain level of qualification, and for a nurse to possess a certain level of qualification, are based on real-life data. Nurse overtime wages depend on their level of qualification and are based on data of the Public Employment Service Austria (AMS). Besides, some variation in the wages of the nurses of a single qualification category is introduced to account for differences in their length of service.

Corresponding to current practices of Austrian home care organizations, a working day, which ranges from 6 a.m. to 8 p.m., is split in five time slots of 2–4 hours. Each job is assigned to one of these time slots, which then represents a hard time window for the start time of the job. Both the distribution of jobs over the time slots and the service duration of a job are based on real-life data. The preferred visit time of a job is randomly selected within the hard time window. To introduce the fact that some jobs have to be performed at a specific time (e.g. certain medical treatments), both the start and end of the hard time window of 5 percent of the jobs are set at the preferred visit time. Finally, regarding nurse preferences, it is assumed that for a certain job each qualified nurse has an equal probability of being preferred, moderately preferred or not preferred.

Four types of instances may be distinguished based on the travel cost and travel time matrices used. The first three types are based on the travel time matrices for car and public transportation provided by Hiermann et al. (2015) and are generated using OpenStreetMap. In types one and two, all nurses are assumed to use car transportation or public transportation respectively, while in the third type some nurses use car transportation while others use public transportation. Since no corresponding travel distance matrices are available, distances in kilometers are assumed to be equal to travel times in minutes. Besides, time is discretized on a 5-minute level. In the fourth type of instances, a distance matrix and a travel time matrix for car transportation between actual job locations of a Viennese service provider are used. In this case, distance and travel time are not perfectly correlated and time is discretized on a 1-minute level. All nurses are assumed to use car transportation, as this is the only information available. Travel costs by car are set at 42 eurocents per kilometer, while it is assumed that no operational costs are incurred when using public transportation (assuming that nurses have a yearly ticket). The number of instances of the third type (mixed mode) is

Table 3
Computation times on small problem instances using the ϵ-constraint method (in minutes).

Instance	Jobs	Base	Y	Y_{BP}	IN1	IN2	IN3	IN4	IN5	IN6	All	Sel.
1	10	0	0	0	0	0	0	0	0	0	0	0
2	10	0	0	0	0	0	0	0	0	0	0	0
3	10	0	0	0	0	0	0	0	0	0	0	0
4	10	0	0	0	0	0	0	0	0	0	0	0
5	10	0	0	0	0	0	0	0	0	0	0	0
6	15	0	0	0	0	0	0	0	0	0	0	0
7	15	0	0	0	0	0	0	0	0	0	0	0
8	15	3	3	2	2	3	3	3	2	3	2	2
9	15	12	10	11	10	12	10	8	12	9	11	8
10	15	2	1	1	1	1	1	1	1	1	1	0
11	20	45	31	25	28	37	34	27	16	14	11	20
12	20	23	12	13	13	7	10	8	19	14	5	6
13	20	2270	1500	871	2582	2322	1930	1238	2667	2337	685	642
14	20	3	2	3	2	3	3	2	3	2	2	2
15	20	41	25	25	25	38	34	30	20	16	18	24
16	20	121	96	133	158	92	103	169	146	100	73	132
17	20	524	450	443	375	316	389	290	495	442	244	278
18	20	30	26	36	43	38	41	34	46	46	14	16
19	20	25	17	35	33	44	30	40	40	35	16	12
20	20	–	–	–	–	–	–	–	–	–	1197	1484
21	25	–	–	–	–	–	–	–	–	–	–	–
22	25	–	–	–	–	–	–	–	–	–	566	382
23	25	–	–	–	–	–	–	–	–	–	–	–
24	25	–	–	–	–	–	–	–	–	–	1532	1338
25	25	–	–	–	–	–	–	–	–	–	–	–
26	25	–	–	–	–	–	–	–	–	–	–	–
27	25	–	–	–	–	–	–	–	–	–	334	351
28	25	–	–	–	–	–	–	–	–	–	–	–
29	25	–	–	–	–	–	–	–	–	–	–	–
30	25	–	–	–	–	–	–	–	–	–	–	–
Avg_{1-19}		163	114	84	172	153	136	97	182	159	57	60

twice that of the other instances, as we believe these instances to be the most realistic and hardest to solve.

Finally, since in practice a patient may have several service requests per day, up to four jobs may be required at the same physical location.

5.2. Results of ϵ-constraint method

To assess the quality of the proposed metaheuristic, Pareto-optimal solutions for small problem instances are generated by embedding the model described in Section 2.1 into the well-known ϵ-constraint scheme (Laumanns, Thiele, & Zitzler, 2006).

Table 3 gives an overview of the computation times in minutes required to solve the small problem instances. Nineteen instances of up to 20 jobs could be solved using the base model within the runtime limit. Most of the enhancements introduced in Section 2.3 reduce average computation times on these instances when applied individually. We also consider applying all enhancements simultaneously ("All") and a version where we only apply the enhancements which reduce overall computation times when applied individually ("Sel"). In both cases, computation times are reduced by more than 60 percent on average compared to the base model and some instances up to 25 jobs can be solved as well. Note that omitting the "bad" enhancements does not give an improvement compared to keeping them all. However, computation times also indicate that solving the problem to optimality for instances of realistic size would be troublesome and hence the use of a heuristic method is justified.

5.3. Results of metaheuristic method

Several quality indicators have been proposed in the literature to evaluate approximations of the Pareto frontier generated by heuristic solution procedures (Knowles, Thiele, & Zitzler, 2006; Zitzler, Thiele, Laumanns, Fonseca, & Grunert da Fonseca, 2003). In this paper, two well-known quality indicators (hypervolume and multiplicative unary epsilon) are used to compare our heuristic results with optimal Pareto fronts and to evaluate different algorithmic designs and parameter settings. For both indicators, the reference set R is equal to the optimal Pareto front if known. Otherwise R is approximated by taking the union of all solutions obtained by any of the experiments during algorithm design and testing, and removing dominated solutions. These reference sets are provided online.

The hypervolume indicator ($I_H(A)$), introduced by Zitzler and Thiele (1999), measures the portion of the objective space that is weakly dominated by an approximation set A. Normalized objective values are used and hence the reference point is (1,1). To allow a meaningful aggregation over all instances, hypervolume results of an approximation set A are presented as the fraction of the hypervolume value of the reference set R that is covered by set A ($I_H^{fr}(A) = I_H(A)/I_H(R)$). High values are preferable. The multiplicative unary epsilon value ($I_\epsilon(A)$) (Zitzler et al., 2003) gives the minimum number ϵ by which each point in the reference set R should be multiplied such that the resulting approximation set is weakly dominated by approximation set A. Low values are preferable. Since one of the objective values for some solutions may be zero, objective values are normalized between 1 and 2 instead of between 0 and 1.

The base version of our algorithm ($v0$) as presented in Section 4 uses the following initial parameter settings. The number of jobs selected to be removed q is distributed uniformly between $\max(2, 0.1 \cdot I)$ and $\min(50, 0.6 \cdot I)$. Both absolute and relative bounds are applied to allow a very large removal rate for small instances while ensuring acceptable computation times for large instances. For the removal operators, the randomness parameter P is set to 5. Maximum noise levels for travel costs (travel time and route duration) is 10 percent of the average travel cost (travel time) in the network. Maximum noise levels for overtime costs are 25 percent of the average of the maximum overtime cost over all nurses. Finally, the probabilities for adding (subtracting) 0.5 and 1 to (from) $incon_i$ are 0.1 and 0.05 respectively. The algorithm is run for 2 million iterations (large time limit), while results after 200,000 iterations are reported as well

Table 4
Average results of different algorithm configurations.

Configuration	Small time limit				Large time limit			
	$I_H^{fr}(A)$	$I_\epsilon(A)$	# Solutions		$I_H^{fr}(A)$	$I_\epsilon(A)$	# Solutions	
			Unique	All			Unique	All
v0	0.8889	1.0712	88	107	0.9450	1.0433	95	166
v1	0.9063	1.0639	93	128	0.9607	1.0339	100	290
v2	0.8985	1.0678	88	109	0.9534	1.0398	95	186
v3	0.8892	1.0724	89	108	0.9472	1.0408	96	167
v4	0.8913	1.0718	88	108	0.9466	1.0413	95	178
v5	0.8958	1.0703	89	109	0.9505	1.0405	95	181
v6	0.9215	1.0571	94	151	0.9719	1.0290	100	393

Table 5
Trade-off analysis.

Instances	Minimum cost solution								Minimum inconvenience solution							
	Cost/job	Inconvenience/job							Cost/job	Inconvenience/job						
	Avg.	Avg.	St.dev	0	1	2	3	4	Avg.	Avg.	St.dev	0	1	2	3	4
All	2.10	1.49	1.28	0.24	0.27	0.30	0.12	0.06	6.54	0.48	0.47	0.69	0.18	0.10	0.02	0.01
Small	4.26	1.76	1.38	0.17	0.27	0.31	0.15	0.11	7.53	1.05	0.99	0.38	0.30	0.24	0.06	0.02
Medium	1.39	1.41	1.26	0.27	0.27	0.30	0.12	0.05	6.72	0.28	0.31	0.78	0.17	0.04	0.01	0.00
Large	0.66	1.30	1.21	0.29	0.29	0.28	0.10	0.04	5.36	0.11	0.13	0.91	0.08	0.01	0.00	0.00
Car	2.41	1.49	1.39	0.26	0.27	0.27	0.12	0.08	6.94	0.39	0.45	0.76	0.12	0.09	0.01	0.01
Public	2.44	1.46	1.28	0.26	0.26	0.31	0.11	0.06	6.89	0.64	0.58	0.59	0.23	0.14	0.02	0.01
Mix	1.79	1.47	1.23	0.24	0.28	0.32	0.10	0.06	6.68	0.48	0.45	0.68	0.20	0.09	0.03	0.01
Car (own)	2.08	1.58	1.29	0.21	0.28	0.28	0.16	0.06	5.51	0.40	0.43	0.74	0.15	0.08	0.03	0.00

(small time limit). Although the former results are generally considerably better than the latter, the latter results represent already good solutions obtained in relatively small computation times (60 seconds on average, compared to 592 seconds for 2 million iterations). Average results over five runs on all instances are reported in the first line of Table 4. Columns two and three indicate the average indicator values. The number of non-dominated solutions with unique objective values and the total number of non-dominated solutions (some having the same objective values but different routing solutions) are presented in columns four and five respectively.

Various configurations of our algorithm have been tested to find good parameter settings and to identify non-contributing components of the algorithm. These configurations make use of different parameters for generating q, different values for P, different maximum noise levels, no noise at all, only a subset of the removal and insertion heuristics or an adaptive version of LNS. To allow a fair comparison of the different configurations, for each instance the algorithm is run for the average amount of computation time required by the base version. In this paper, only those configurations that improve at least three out of four quality indicators ($I_H^{fr}(A)$ and $I_\epsilon(A)$, for both time limits) are reported in Table 4. These configurations include:

- changing the lower bound on the interval for q to $\max(2, 0.05 \cdot I)$ (v1),
- changing the upper bound on the interval for q to $\max(40, 0.60 \cdot I)$ (v2),
- changing the value of P to 4 (v3),
- not applying the *related* removal operator when minimizing costs (v4),
- not applying the *basic greedy* insertion operator for both objectives (v5),
- a combination of configurations v1 to v5 (v6).

The final configuration clearly provides the best results. Detailed results for this configuration are provided in Appendix B. Note that the algorithm provides the complete optimal Pareto front for 20 out of the 23 instances for which this front is known ($I_H^{fr}(A) = I_\epsilon(A) = 1$). For the other instances only small differences with the optimal front

exist, except for instance 27. The rather bad indicator values for this instance may be caused by reducing the number of operators and the smaller (bounds on the) removal rate q, changes which appear to be highly beneficial for solution quality on larger instances. Furthermore, the fact that the number of solutions in the optimal front is rather low for this instance (9), makes that not finding one of these solutions may have a comparably large effect on the indicator values.

Table 4 already indicates that a set of solutions may contain several solutions with the same objective values but different routes. Vice versa, due to the bi-objective nature of the scheduling problem, a set of routes may result in multiple non-dominated solutions. In fact, the set of non-dominated solutions with unique routes on average only represents 82 percent of the total number of non-dominated solutions found by the algorithm.

5.4. Trade-off analysis

The trade-off between both objectives is analyzed using five runs of the best settings of the metaheuristic algorithm (v6). For this purpose, several subsets of the 90 instances are considered. A distinction is made between small instances ($I \leq 25$), medium-sized instances ($50 \leq I \leq 150$) and large instances ($I \geq 200$). Besides, a distinction is made based on the considered mode(s) of transportation and travel cost/time data (car, public, mix, car (own data)). A difficulty with analyzing the trade-off is the fact that for some solutions one of the objective values may be zero, thereby making it impossible to express for example the increase in costs when reducing client inconvenience as a percentage of the minimum cost.

Information on the two extreme solutions in an approximation set, the one with minimum cost and the one with minimum inconvenience, is presented in Table 5. For each set of instances, average values over five runs on all instances are reported. The first two columns indicate the average cost per job in euros and the average level of inconvenience for a job in the solution. Values per job instead of total values are used to allow a fairer comparison between (sets of) instances. However, comparison between sets of instances
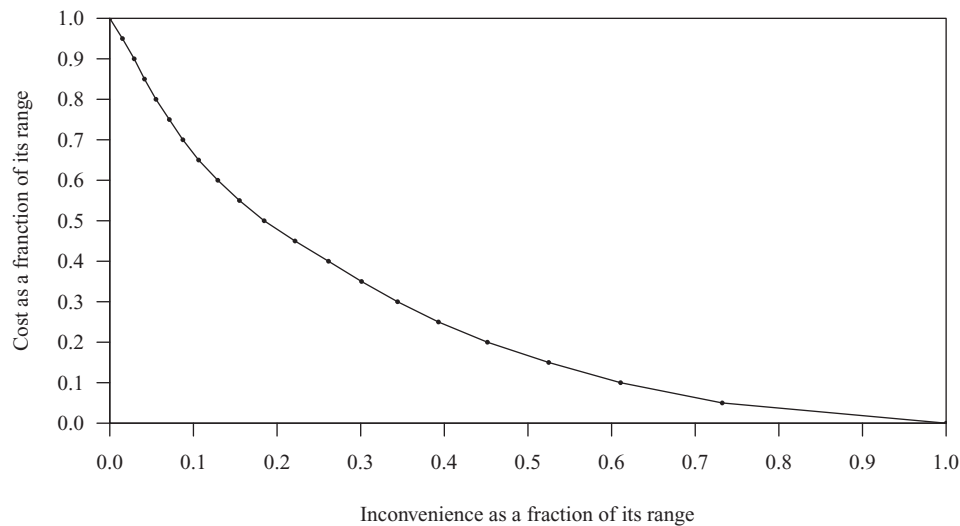
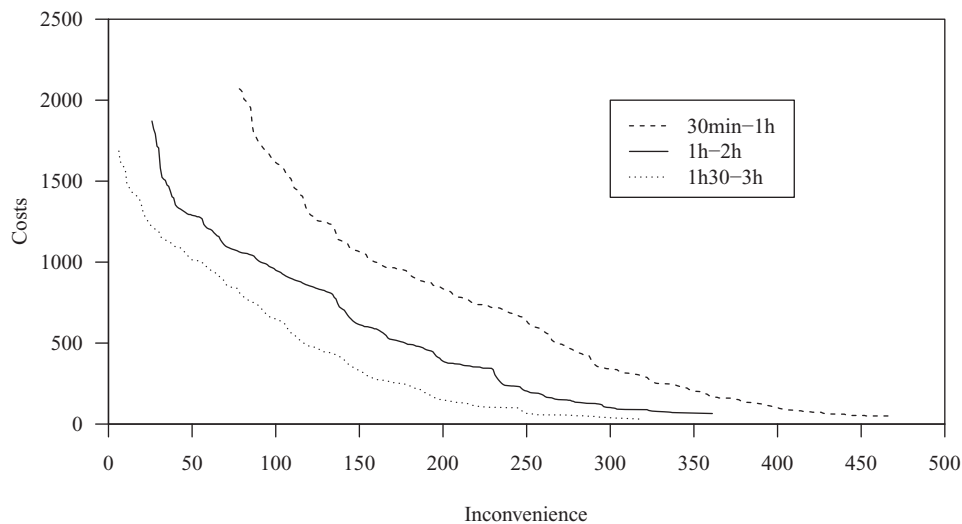**Fig. 1.** Average shape of the trade-off curve.

Fig. 2. Effect of the width of the soft time windows.

should still be made with care as the underlying network information differs from instance to instance. Since inconvenience is clearly attributable to a specific job and the inconvenience level may only take integer values from zero to four, some additional information on the distribution of total inconvenience over the jobs is presented as well. The third column indicates the standard deviation from the average, while the other columns indicate the fraction of jobs having a specific inconvenience level, averaged over all runs and instances.

Results indicate that on average a considerable difference between the minimum cost and minimum inconvenience solution exists in terms of both objectives. Hence, the decision of service providers on which service level to offer to their clients has a large effect on operating costs and this decision should be made carefully. While the mode of transportation does not seem to have a large effect on the extreme solutions, the cost and inconvenience per job seem to decrease considerably when the number of jobs increases, indicating economies of scale for larger service providers. Looking at the distribution of inconvenience over the jobs for the minimum inconvenience solution, it is clear that most jobs have a small level of inconvenience (0 or 1). Although the objective is to minimize total inconvenience, without imposing constraints on the inconvenience of a specific job, only very few jobs seem to suffer from a very high level of inconvenience (3 or 4). Even for the minimum cost solution, on aver-

age jobs still have a relatively low level of inconvenience (1.49), with most jobs having levels of 0, 1 or 2.

More information on the trade-off may be found in Fig. 1 which indicates the average shape of the trade-off curve between the two extreme solutions over all runs and instances. The figure is constructed as follows. For each run on each instance, the range of total costs is defined as the difference between total costs in the minimum inconvenience solution and the minimum cost solution. This range indicates the additional costs a service provider might incur in order to improve the offered service level to his clients. Similarly the range of inconvenience is defined as the reduction in inconvenience that may be achieved by incurring these additional costs. Next, starting from the minimum cost, for every increase in costs equal to a multiple of 5 percent of the total cost range, the corresponding inconvenience level is found. This is done by calculating the total cost level, looking for the two solutions in the front which encompass this cost level and linearly interpolating to find the inconvenience level which corresponds to the cost level. The difference between this inconvenience level and the minimum inconvenience level is then expressed as a fraction of the range of inconvenience. Fig. 1 shows the average fractions over all runs and instances. It indicates which fraction of the total possible reduction in inconvenience is achieved on average when increasing costs with a certain fraction of its range. Since fractions of the ranges

Table A.1
Overview of instances.

Instance	Jobs	Patients	Nurses FT	PT	Modes[1]	Inst.	Jobs	Pat.	Nurses FT	PT	Modes	Inst.	Jobs	Pat.	Nurses FT	PT	Modes
1	10	6	2	0	0	31	50	33	6	8	0	61	200	117	21	38	0
2	10	6	2	2	1	32	50	25	6	8	1	62	200	112	21	38	1
3	10	6	2	0	2	33	50	27	6	8	2	63	200	117	21	38	2
4	10	6	2	0	3	34	50	34	6	8	3	64	200	110	21	38	3
5	10	6	2	2	2	35	50	27	6	8	2	65	200	115	21	38	2
6	15	6	2	2	0	36	50	32	6	8	0	66	200	106	21	38	0
7	15	6	3	2	1	37	50	25	6	8	1	67	200	108	21	38	1
8	15	8	3	2	2	38	50	31	6	8	2	68	200	106	21	38	2
9	15	7	2	2	3	39	50	31	6	8	3	69	200	113	21	38	3
10	15	9	3	2	2	40	50	28	6	8	2	70	200	115	21	38	2
11	20	13	3	2	0	41	100	56	11	18	0	71	250	141	26	48	0
12	20	10	3	2	1	42	100	56	11	18	1	72	250	142	26	48	1
13	20	14	3	2	2	43	100	53	11	18	2	73	250	150	26	48	2
14	20	14	3	2	3	44	100	55	11	18	3	74	250	152	26	48	3
15	20	12	3	2	2	45	100	60	11	18	2	75	250	135	26	48	2
16	20	13	3	2	0	46	100	53	11	18	0	76	250	149	26	48	0
17	20	14	3	2	1	47	100	63	11	18	1	77	250	135	26	48	1
18	20	11	3	2	2	48	100	50	11	18	2	78	250	140	26	48	2
19	20	11	3	2	3	49	100	58	11	18	3	79	250	149	26	48	3
20	20	11	3	2	2	50	100	63	11	18	2	80	250	154	26	48	2
21	25	14	3	4	0	51	150	90	16	28	0	81	300	170	31	58	0
22	25	13	3	4	1	52	150	89	16	28	1	82	300	164	31	58	1
23	25	16	3	4	2	53	150	83	16	28	2	83	300	164	31	58	2
24	25	12	3	4	3	54	150	90	16	28	3	84	300	168	31	58	3
25	25	16	3	4	2	55	150	90	16	28	2	85	300	167	31	58	2
26	25	13	3	4	0	56	150	85	16	28	0	86	300	182	31	58	0
27	25	15	3	4	1	57	150	85	16	28	1	87	300	180	31	58	1
28	25	16	3	4	2	58	150	91	16	28	2	88	300	165	31	58	2
29	25	13	3	4	3	59	150	87	16	28	3	89	300	171	31	58	3
30	25	13	3	4	2	60	150	99	16	28	2	90	300	167	31	58	2

[1] 0: Car, 1: Public, 2: Mixed, 3: Car (own data).

Table A.2
Information on nurses and qualifications.

		Nurse qualifications						Probability
		0	1	2	3	4	5	
Job qualifications	0	x	x	x	x			0.05
	1		x	x	x			0.02
	2			x	x	x	x	0.68
	3				x	x	x	0.11
	4					x	x	0.12
	5						x	0.01
Probability		0.01	0.04	0.62	0.09	0.11	0.12	
Min. wage (euro/overtime hour)		20.18	25.45	28.57	27.59	30.98	37.81	
Max. wage (euro/overtime hour)		22.32	28.21	31.88	34.46	37.81	44.64	

of both objectives are used, the actual slope of the trade-off curve will depend on the actual absolute ranges. For example, a large cost range and a small inconvenience range will result in a rather steep curve. Besides, note that for individual instances the trade-off curve might not be convex, i.e. some efficient solutions may be non-supported and cannot be found using a simple weighted objective approach. For service providers mainly focusing on minimizing costs, the curve in Fig. 1 indicates that with a relatively small effort in terms of costs, inconvenience for the clients may already be reduced considerably. An increase in costs of respectively 5 or 10 percent of its range will already result in a reduction of inconvenience of 27 or 39 percent of its range. As can be expected, the more one moves towards the minimum inconvenience solution, the more costly it becomes to reduce inconvenience even further.

Finally, the effect of the width of the soft time windows around the preferred visit time is analyzed. Service providers may set the width of these time windows in accordance with their view on client expectations. The less clients are prepared to allow a deviation from their preferred visit time, the narrower these time windows should be set. For all experiments discussed above, the soft time windows had a

Table A.3
Information on time parameters.

Time slots	Probability	Service duration (in hours)			
		Avg.	St.dev.	Min.	Max.
6:00−7:59	0.20	1.1015	0.3715	0.25	2.5
8:00−10:59	0.34	1.5188	0.8105	0.50	4
11:00−12.59	0.28	1.0093	0.5008	0.50	4
13:00−15:59	0.04	1.1027	0.5073	0.50	3
16:00 −20:00	0.14	0.7209	0.1745	0.50	2

Table A.4
Jobs per patient.

Jobs	1	2	3	4
Probability	0.52	0.26	0.17	0.05

width of 1 and 2 hours, centered around the preferred visit time. The effect of widening (to 1 hour 30 minutes and 3 hours) and narrowing (to 30 minutes and 1 hour) these time windows is shown in Fig. 2. It depicts the non-dominated solutions found by the algorithm for each

Table B.1
Detailed results of algorithm configuration *v6*.

Configuration	First time limit						Second time limit					
	$I_H^{fr}(A)$	$I_\epsilon(A)$	# Solutions		Runtime	# Iterations	$I_H^{fr}(A)$	$I_\epsilon(A)$	# Solutions		Runtime	# Iterations
			Unique	All					Unique	All		
1[1]	1.0000	1.0000	7	7	4	322371	1.0000	1.0000	7	7	22	2691070
2[1]	1.0000	1.0000	5	5	3	282485	1.0000	1.0000	5	5	21	2694370
3[1]	1.0000	1.0000	3	3	3	291106	1.0000	1.0000	3	3	20	2703950
4[1]	1.0000	1.0000	10	10	4	341821	1.0000	1.0000	10	10	24	2707320
5[1]	1.0000	1.0000	3	4	3	285050	1.0000	1.0000	3	4	21	2618540
6[1]	1.0000	1.0000	11	11	4	255984	1.0000	1.0000	11	11	31	2645000
7[1]	1.0000	1.0000	4	4	4	278697	1.0000	1.0000	4	4	28	2721530
8[1]	1.0000	1.1005	3	3	4	258437	1.0000	1.0000	3	3	31	2660400
9[1]	1.0000	1.0000	8	8	4	246401	1.0000	1.0000	8	8	32	2710920
10[1]	1.0000	1.0000	8	16	4	243982	1.0000	1.0000	8	16	32	2667950
11[1]	1.0000	1.0000	7	14	6	278093	1.0000	1.0000	7	14	44	2653590
12[1]	1.0000	1.0000	4	4	4	220984	1.0000	1.0000	4	4	36	2751050
13[1]	1.0000	1.0375	13	29	5	229217	1.0000	1.0000	14	31	44	2663390
14[1]	0.9995	1.0023	12	18	5	247119	1.0000	1.0000	12	17	41	2743380
15[1]	1.0000	1.0000	5	13	5	245938	1.0000	1.0000	5	13	40	2700010
16[1]	0.9967	1.0166	12	30	5	232229	0.9976	1.0133	12	30	43	2679460
17[1]	1.0000	1.0000	4	4	5	268351	1.0000	1.0000	4	4	38	2715520
18[1]	1.0000	1.0000	12	19	5	233234	1.0000	1.0000	12	19	42	2705350
19[1]	0.9990	1.0066	7	7	6	286213	1.0000	1.0000	7	7	43	2757600
20[1]	0.9841	1.0432	9	11	5	251434	1.0000	1.0000	10	14	39	2729790
21	0.9904	1.0318	17	19	7	229940	0.9995	1.0054	19	21	63	2682040
22[1]	0.9946	1.0348	23	53	6	242204	0.9959	1.0261	23	53	51	2774820
23	0.9577	1.0353	9	19	7	259000	1.0000	1.0000	9	20	56	2693320
24[1]	1.0000	1.0000	7	7	6	247483	1.0000	1.0000	7	7	51	2783750
25	0.9924	1.0257	19	49	6	218788	0.9986	1.0052	19	43	56	2700240
26	0.9881	1.0329	21	29	8	252751	0.9969	1.0255	23	35	66	2736450
27[1]	0.8364	1.3005	6	6	6	252593	0.9343	1.1394	8	8	49	2843050
28	0.9972	1.0194	18	52	7	253171	1.0000	1.0000	18	51	57	2736360
29	0.9919	1.0284	18	44	7	232890	0.9962	1.0234	19	51	63	2770440
30	1.0000	1.0000	14	22	6	233665	1.0000	1.0000	14	22	53	2734000
31	0.9351	1.0810	37	476	19	217197	0.9544	1.0583	37	559	185	2725800
32	0.8974	1.0690	39	60	16	222906	0.9513	1.0517	37	60	152	2794580
33	0.9416	1.0592	50	65	17	215487	0.9648	1.0380	57	93	164	2705710
34	0.9355	1.0604	49	67	18	223049	0.9605	1.0474	54	68	170	2884650
35	0.9451	1.0492	36	76	18	221306	0.9628	1.0419	37	79	169	2840250
36	0.9712	1.0366	47	130	18	213989	0.9798	1.0324	51	143	177	2827630
37	0.9335	1.0626	23	26	15	228768	0.9689	1.0437	27	31	138	2839860
38	0.9613	1.0390	43	78	16	220366	0.9801	1.0281	47	82	153	2815620
39	0.9523	1.0505	49	140	19	228582	0.9831	1.0269	51	142	176	2907700
40	0.9667	1.0337	47	132	17	220942	0.9781	1.0270	51	158	161	2791240
41	0.9192	1.0580	78	299	51	274777	0.9590	1.0380	88	1240	504	2814610
42	0.9000	1.0732	54	126	43	275827	0.9616	1.0442	56	315	417	2922740
43	0.8688	1.0912	73	149	47	277187	0.9310	1.0611	77	343	454	2809980
44	0.8936	1.0644	91	110	48	277917	0.9564	1.0379	108	166	469	2723040
45	0.8515	1.1125	76	183	43	277093	0.9268	1.0629	91	358	421	2696050
46	0.9237	1.0496	91	191	50	274703	0.9682	1.0280	99	246	491	2706940
47	0.8231	1.1106	60	243	40	274956	0.9015	1.0749	67	577	397	2717750
48	0.9181	1.0590	72	164	49	272636	0.9672	1.0306	80	410	481	2696040
49	0.9113	1.0542	103	160	51	274866	0.9629	1.0310	119	310	501	2683230
50	0.8245	1.1338	63	335	43	282215	0.8896	1.0880	76	1526	416	2706030
51	0.9309	1.0478	115	231	76	275950	0.9690	1.0302	126	868	747	2694410
52	0.8925	1.0705	126	192	62	271299	0.9577	1.0436	125	638	604	2625660
53	0.8437	1.0997	103	212	59	268968	0.9430	1.0612	116	860	589	2639600
54	0.9298	1.0523	134	284	78	277235	0.9690	1.0330	152	993	767	2712780
55	0.8671	1.0776	117	184	66	273112	0.9495	1.0439	123	537	645	2656050
56	0.9258	1.0494	124	251	77	274260	0.9719	1.0294	131	1021	762	2724560
57	0.8315	1.1156	102	174	58	271762	0.9426	1.0529	119	434	571	2665890
58	0.8817	1.0694	126	277	60	268710	0.9629	1.0315	126	720	591	2656460
59	0.8346	1.0880	137	182	66	279500	0.9436	1.0424	160	320	643	2759500
60	0.8555	1.0904	100	179	62	273971	0.9527	1.0469	115	391	602	2665460
61	0.9322	1.0494	142	346	100	276093	0.9546	1.0371	149	1999	987	2738750
62	0.8354	1.1009	134	189	80	274146	0.9459	1.0520	150	461	793	2688050
63	0.8822	1.0737	152	228	87	270057	0.9609	1.0354	156	542	862	2686490
64	0.8869	1.0657	196	232	96	272925	0.9713	1.0286	222	544	953	2736870
65	0.8815	1.0672	154	282	88	271487	0.9580	1.0370	152	552	873	2697460
66	0.9161	1.0526	136	212	101	279382	0.9707	1.0284	147	922	995	2745010
67	0.8005	1.1155	132	212	83	274922	0.9382	1.0473	156	505	822	2712940
68	0.8439	1.0841	164	269	83	270783	0.9421	1.0480	162	849	825	2692670
69	0.9237	1.0488	160	203	105	278501	0.9760	1.0223	194	639	1042	2773290
70	0.8404	1.0998	155	220	81	273566	0.9481	1.0431	160	341	802	2698750
71	0.8818	1.0663	203	304	134	278977	0.9536	1.0363	221	1685	1324	2757720
72	0.8416	1.0992	177	245	104	275415	0.9570	1.0485	200	680	1032	2705170

Table B.1 (*continued*)

Configuration	First time limit						Second time limit					
	$I_H^{fr}(A)$	$I_\epsilon(A)$	# Solutions		Runtime	# Iterations	$I_H^{fr}(A)$	$I_\epsilon(A)$	# Solutions		Runtime	# Iterations
			Unique	All					Unique	All		
73	0.8348	1.0812	178	217	114	275871	0.9641	1.0339	201	455	1132	2710080
74	0.9219	1.0663	180	266	149	283546	0.9751	1.0282	209	1586	1466	2829770
75	0.8108	1.1076	195	235	114	275883	0.9537	1.0455	217	457	1135	2722950
76	0.9088	1.0528	199	254	139	278235	0.9788	1.0210	202	722	1372	2769220
77	0.8238	1.0978	176	228	105	276788	0.9584	1.0400	181	352	1046	2701120
78	0.8575	1.0869	217	256	121	275645	0.9633	1.0310	221	518	1201	2702410
79	0.9070	1.0683	207	247	143	281027	0.9722	1.0328	245	528	1416	2790080
80	0.8838	1.0772	188	221	121	272210	0.9631	1.0372	177	441	1195	2693270
81	0.9092	1.0626	194	254	212	289459	0.9754	1.0320	200	614	2100	2869860
82	0.7956	1.1089	195	240	139	285936	0.9504	1.0445	226	552	1393	2828100
83	0.8598	1.0771	243	291	172	280946	0.9618	1.0377	242	666	1719	2828220
84	0.8853	1.0669	302	327	204	286940	0.9698	1.0284	313	555	2015	2844880
85	0.8281	1.0882	236	278	164	284656	0.9479	1.0401	265	558	1629	2805600
86	0.9074	1.0674	185	228	211	289169	0.9778	1.0288	179	550	2093	2881800
87	0.8825	1.0743	218	272	154	275307	0.9762	1.0266	205	393	1540	2785550
88	0.8685	1.0858	238	291	171	283158	0.9659	1.0377	235	452	1702	2787670
89	0.9248	1.0516	253	307	220	287101	0.9781	1.0248	282	1250	2194	2898160
90	0.8676	1.0719	242	299	169	285335	0.9705	1.0272	245	794	1665	2793740

[1] Pareto-optimal solutions are known.

of these settings for one of the largest instances (instance 90). Clearly, when the soft time windows are narrower it becomes more costly to offer a similar service level and hence the trade-off curve shifts to the upper right corner.

6. Conclusions and future research

Demand for home care services in western countries is increasing due to demographic changes in terms of a continuously aging population. Home care providers are faced with the need for tools to support and optimize their operational routing and scheduling to cope with this increasing demand. This optimization problem consists of assigning jobs to nurses and constructing efficient routes and schedules for the nurses. It has been studied extensively in the past years, often inspired by real-life applications.

Service providers are confronted with multiple, often conflicting, objectives in this process. On the one hand the objective is to minimize their operating costs while on the other hand they want to offer a high service level to their clients by taking into account their preferences. Current planning models either use weighted objective functions or hard constraints to incorporate the latter aspect, thereby masking or ignoring the trade-off between both objectives. Since the trade-off between cost and client convenience is an important consideration for service providers, this paper is the first to shed some light on this trade-off relationship by modeling the home care routing and scheduling problem as a bi-objective problem.

A definition of the problem is presented, together with a mixed integer problem formulation and some valid inequalities to strengthen this formulation. The problem takes into account qualifications, working regulations and overtime costs of the nurses, travel costs depending on the mode of transportation, hard time windows, and client preferences on visit times and nurses. A distinguishing characteristic of the problem is that the scheduling problem for a single route is a bi-objective problem in itself, thereby complicating the problem considerably.

Small problem instances are solved by applying the ϵ-constraint solution framework. In order to solve problem instances of realistic size, a metaheuristic algorithm is proposed. This algorithm embeds a large neighborhood search heuristic in the multi-directional local search framework. A set of benchmark instances is generated using real-life data, and computational experiments of different parameter settings of the metaheuristic algorithm are presented. A comparison

with exact solutions on small instances shows that the algorithm adequately solves the problem under study. An analysis of the results reveals that service providers face a considerable trade-off between costs and client convenience. However, starting from a minimum cost solution, the average service level offered to clients may already be improved drastically with a relatively small fraction of additional costs, e.g. on average an increase in costs of respectively 5 or 10 percent of its range already results in a reduction of inconvenience of 27 or 39 percent of its range.

As this paper is the first to study the home care scheduling problem from a bi- or multi-objective perspective, many opportunities for future research exist. More sophisticated exact solution approaches (e.g. Branch-and-Cut(-and-Price)) may be developed to assess the quality of heuristic procedures for larger instances. Furthermore, the problem and the MDLS-based metaheuristic can easily be extended with additional objectives. Nurse convenience may for example be included to address the preferences of nurses regarding working times and the clients to visit. Finally, the problem may be extended with additional real-life aspects such as temporal dependencies between jobs, dynamic aspects of visits and travel times, and a longer planning horizon.

Acknowledgments

Appendix A. Info on problem instance generation

In this appendix detailed information on the parameters used to generate the problem instances is provided. Table A.1 gives an overview of the instances, indicating for each instance the number of jobs, the number of patient locations, the number of full-time (FT) and part-time (PT) nurses, and the transportation modes considered. In Table A.2 the probability for a job to require a certain level of qualification and for a nurse to possess a certain level of qualification are shown, together with a compatibility matrix. Nurse wages are distributed uniformly between the bounds provided in the last two lines of the table. Information on the time-related parameters is provided

in Table A.3. The second column shows the probability for a job to be assigned to a time slot. For each time slot, the service duration of a job is generated according to a normal distribution with a time slot-specific average and standard deviation, although a hard minimum and maximum are applied to avoid unrealistic values. Finally, Table A.4 indicates the probability for a patient to request several jobs on a single day, thereby reducing the number of physical locations in the network.

Appendix B. Detailed results

See Table B.1 for detailed results.

References

Akjiratikarl, C., Yenradee, P., & Drake, P. R. (2007). PSO-based algorithm for home care worker scheduling in the UK. *Computers & Industrial Engineering, 53*(4), 559–583.

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

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

Borsani, V., Matta, A., Beschi, G., & Sommaruga, F. (2006). A home care scheduling model for human resources. In *Service systems and service management, 2006 international conference on: vol. 1* (pp. 449–454).

Campbell, A. M., & Savelsberg, M. (2004). Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation Science, 38*(3), 369–378.

Cappanera, P., & Scutellà, M. G. (2013). Joint assignment, scheduling and routing models to home care optimization: a pattern based approach. TR-13-05. Dipartimento di Informatica, Universitá di Pisa.

Castillo-Salazar, J., Landa-Silva, D., & Qu, R. (2015). Workforce scheduling and routing problems: literature survey and computational study. *Annals of Operations Research, Forthcoming.* doi:10.1007/s10479-014-1687-2.

Cheng, E., & Rich, J. L. (1998). A home health care routing and scheduling problem. TR98-04. Rice University, USA.

De Bruecker, P., Van den Bergh, J., Beliën, J., & Demeulemeester, E. (2015). Workforce planning incorporating skills: State of the art. *European Journal of Operational Research, 243*(1), 1–16.

Ehrgott, M. (2005). *Multicriteria optimization* (2nd ed). Berlin: Springer.

Ehrgott, M., & Gandibleux, X. (2002). Multiobjective combinatorial optimization. In M. Ehrgott, & X. Gandibleux (Eds.), *International Series in Operations Research and Management: Vol. 52. Multiple criteria optimization-state of the art annotated bibliographic surveys* (pp. 369–444). Boston, MA: Kluwer Academic Publishers.

Ehrgott, M., & Gandibleux, X. (2004). Approximative solution methods for multiobjective combinatorial optimization. *TOP, 12*(1), 1–90.

EuropeanCommission. (2007). Europe's demographic future: Facts and figures on challenges and opportunities. ISBN 978-92-79-07043-3.

Eveborn, P., Flisberg, P., & Rönnqvist, M. (2006). LAPS CARE – an operational system for staff planning of home care. *European Journal of Operational Research, 171*(3), 962–976.

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

Gamst, M., & Jensen, T. (2012). A branch-and-price algorithm for the long-term home care scheduling problem. In D. Klatte, H.-J. Lüthi, & K. Schmedders (Eds.), *Operations research proceedings 2011* (pp. 483–488). Berlin Heidelberg: Springer.

Gutiérrez, E. V., Gutiérrez, V., & Vidal, C. J. (2013). Home health care logistics management: Framework and research perspectives. *International Journal of Industrial Engineering and Management, 4*(3), 173–182.

Hashimoto, H., Yagiura, M., Imahori, S., & Ibaraki, T. (2013). Recent progress of local search in handling the time window constraints of the vehicle routing problem. *Annals of Operations Research, 204*, 171–187.

Hiermann, G., Prandtstetter, M., Rendl, A., Puchinger, J., & Raidl, G. R. (2015). Metaheuristics for solving a multimodal home-healthcare scheduling problem. *Central European Journal of Operations Research, 23*(1), 89–113.

Kergosien, Y., Lenté, C., & Billaut, J.-C. (2009). Home health care problem: An extended multiple traveling salesman problem. In *4th multidisciplinary international conference on scheduling: Theory and applications (MISTA'09), Dublin* (pp. 10–12).

Kindervater, G. A. P., & Savelsbergh, M. W. P. (1997). Vehicle routing: Handling edge exchanges. In E. Aarts, & J. K. Lenstra (Eds.), *Local search in combinatorial optimization* (pp. 337–360). Wiley.

Knowles, J. D., Thiele, L., & Zitzler, E. (2006). A tutorial on the performance assessment of stochastic multiobjective optimizers. *Technical Report TIK-Report No. 214, Computer Engineering and Networks Laboratory*. ETH Zurich, Zurich.

Koeleman, P., Bhulai, S., & van Meersbergen, M. (2012). Optimal patient and personnel scheduling policies for care-at-home service facilities. *European Journal of Operational Research, 219*(3), 557–563.

Laumanns, M., Thiele, L., & Zitzler, E. (2006). An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research, 169*(3), 932–942.

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

Maya Duque, P. A., Castro, M., Sörensen, K., & Goos, P. (2015). Home care service planning. the case of Landelijke Thuiszorg. *European Journal of Operational Research, 243*(1), 292–301.

Nickel, S., Schröder, M., & Steeg, J. (2012). Mid-term and short-term planning support for home health care services. *European Journal of Operational Research, 219*(3), 574–587.

Nowak, M., Hewitt, M., & Nataraj, N. (2013). Loyola University Chicago.

Pisinger, D., & Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research, 34*(8), 2403–2435.

Pisinger, D., & Ropke, S. (2010). Large neighborhood search. In M. Gendreau, & J.-Y. Potvin (Eds.), *International Series in Operations Research & Management Science: vol. 146. Handbook of Metaheuristics* (pp. 399–419). US: Springer.

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

Ropke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science, 40*(4), 455–472.

Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In M. Maher, & J.-F. Puget (Eds.), *Lecture Notes in Computer Science: vol. 1520. Principles and Practice of Constraint Programming CP98* (pp. 417–431). Berlin Heidelberg: Springer.

Tarricone, R., & Tsouros, A. D. (Eds.). (2008). *The solid facts: Home care in Europe*. World Health Organization.

Trautsamwieser, A., Gronalt, M., & Hirsch, P. (2011). Securing home health care in times of natural disasters. *OR Spectrum, 33*(3), 787–813.

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.

Tricoire, F. (2012). Multi-directional local search. *Computers & Operations Research, 39*(12), 3089–3101.

Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., & De Boeck, L. (2013). Personnel scheduling: A literature review. *European Journal of Operational Research, 226*(3), 367–385.

Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2015). Timing problems and algorithms: Time decisions for sequences of activities. *Networks, Forthcoming.* doi:10.1002/net.21587.

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation, 3*(4), 257–271.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & Grunert da Fonseca, V. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation, 7*(2), 117–132.