Operations Research for Health Care 16 (2018) 59–71

Contents lists available at ScienceDirect

Operations Research for Health Care

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

A memetic algorithm for a home health care routing and scheduling problem

Jérémy Decerle [a,*], Olivier Grunder [a], Amir Hajjam El Hassani [a], Oussama Barakat [b]

[a] Nanomedicine Lab, Univ. Bourgogne Franche-Comté, UTBM, F-90010 Belfort, France
[b] Nanomedicine Lab, Univ. Bourgogne Franche-Comté, F-25000 Besançon, France

ARTICLE INFO

Article history:
Received 22 May 2017
Accepted 12 January 2018
Available online 1 March 2018

Keywords:
Metaheuristics
Memetic algorithm
Home health care
Temporal constraint
Soft constraints

ABSTRACT

This work addresses a home health care routing and scheduling problem with time window and synchronization constraints. Each patient is associated with a period of availability according to their preferences while some visits may require the presence of two staff members simultaneously, which requires the synchronization of two visits. In this paper, the problem is studied with hard and soft patients time window and synchronization constraints. We developed a mixed integer programming model and a memetic algorithm featuring two original crossover operators. Experiments are conducted on benchmark instances from the literature as well as new instances based on real life data from a home health care provider in France. The results highlight the efficiency of the memetic algorithm since it provides great results while being flexible to the instance type. Indeed, the memetic algorithm is efficient whether the problem is studied with hard or soft time window and synchronization constraints, various caregivers qualification or several home health care offices.

1. Introduction

The vehicle routing and scheduling problem (VRP) is an extensive research area in the operations research field. Vehicle routing and scheduling problem consists in establishing a valid schedule for each vehicle given a fixed number of vehicles available at the depot and a set of customers to be serviced during a specific time window while minimizing, for instance, the total traveling time of the vehicles. Temporal constraints are frequently considered using time window restrictions for vehicles or customers to serve. However, dependencies between vehicles have been less studied. Dependencies occur when two vehicles have to serve the same customer in a specific order or at the same time, which happens in many practical cases including home health care. Consequently, these two constraints have received less attention in the literature when they are considered together. Moreover, they have been mostly studied in the case of hard time window and same time arrival for two dependent vehicles.

Both exact methods and heuristics have been studied on a large variety of problems. Several surveys have been presented such as Bräysy and Gendreau [1] who focused on metaheuristics or Laporte [2] who studied exact and approximate approaches. Generally speaking, the existing literature is now divided into two categories depending on the main optimized criteria: those that minimize costs (work, transport, ...) and others that minimize time (travel, work, ...).

Home health care structures provide care for the elderly or patients with chronic diseases. Different types of cares are performed depending on the need of the patients. Health care teams are often composed of auxiliary nurses and nurses to provide the full range of cares required such as personal cleaning, injections, bandage and much more.

Nowadays, the planning and scheduling of the home health care staff are performed manually by a coordinating nurse. This complex task often requires an extensive time to obtain a valid schedule which respects all the constraints (availability of patients, staff working time window, ...). Since this task is performed every day (for planning the next day or the coming weekend), there is an important axis of improvement in order to improve the quality of the resulting schedule.

Due to the growing demand for home health care as reported by Bertrand [3], organizations providing home health care services are willing to optimize their activities in order to meet the increasing demand. Consequently, research on this problem has appeared by the end of the 20th century focusing on the different variants of the problem as compared by Fikar and Hirsch [4].

An overview of the characteristics and constraints considered in the following cited publications is provided in Table 1.

* Corresponding author.
E-mail addresses: jeremy.decerle@utbm.fr (J. Decerle), olivier.grunder@utbm.fr (O. Grunder), amir.hajjam-el-hassani@utbm.fr (A. Hajjam El Hassani), oussama.barakat@univ-fcomte.fr (O. Barakat).

https://doi.org/10.1016/j.orhc.2018.01.004
2211-6923/

Table 1
Constraints and assumptions considered in the related works on scheduling and routing of home health care service. The care demand can be either deterministic (Det.) or stochastic (Stoch.).

Reference	Nurse TW		Patient TW		Nurse's skills	Continuity of care	Synchronized visits		Care demand	
	Hard	Soft	Hard	Soft			Hard	Soft	Det.	Stoch.
Afifi et al. [5]	X		X				X		X	
Allaoua et al. [6]	X		X		X				X	
Bertels and Fahle [7]		X		X	X				X	
Braekers et al. [8]		X		X	X	X			X	
Bredström and Rönnqvist [9]	X		X		X	X	X		X	
Cheng and Rich [10]	X		X						X	
Eveborn et al. [11]	X		X		X	X	X		X	
Hiermann et al. [12]	X		X		X				X	
Issaoui et al. [13]	X				X				X	
Kergosien et al. [14]	X		X		X		X		X	
Lanzarone and Matta [15]					X	X				X
Liu et al. [16]			X			X			X	
Mankowska et al. [17]			X		X		X		X	
Rasmussen et al. [18]	X		X				X		X	
Shi et al. [19]			X							X
Trautsamwieser and Hirsch [20]		X		X	X	X			X	
Wirnitzer et al. [21]					X	X			X	
Yalçındağ et al. [22]									X	

In the context of the vehicle routing problem applied to the home health care problem, Cheng and Rich [10] consider the vehicle routing problem with time window (VRPTW) including multiple depots. By considering both full-time and part-time nurses, they have optimized the cost related to the working hours. As full-time nurses work overtime and part-time nurses are paid for each hour of work, their objective is to minimize the amount of overtime and part-time work scheduled. The home health care problem with multiple depots has also been studied by Allaoua et al. [6] who have split the problem into two sub-problems: a set-partitioning-like problem and a multi-depot traveling salesman problem with time window (MDTSPTW) for the routing part. They suggest a matheuristic for solving large instances. Some authors have focused their attention on the trade-off between the costs (coming from work and transports) and the patient convenience such as Issaoui et al. [13] and Braekers et al. [8]. In that case, the problem is modeled using multiple objective functions to be solved. In addition, particular attention on the continuity of care has been paid by Wirnitzer et al. [21]. They have sought ways to minimize the number of different nurses assigned to each patient on a monthly planning period. Liu et al. [16] have also taken into account the continuity of care in their mathematical model considering the lunch break requirements.

Another characteristic of the home health care problem is the qualification of the caregivers. Indeed, all care cannot be performed by the same type of caregivers. Therefore, it is necessary to define the qualification of caregivers to provide care. In their work, Hiermann et al. [12] have considered five kinds of caregivers : community service worker, visiting nurse, home-care nurse, advanced home-care nurse and medical nurse.

Moreover, Lanzarone and Matta [15] have proposed a solution to the assignment problem under the continuity of care that minimizes nurses' maximum overtimes. In opposite to Yalçındağ et al. [22] who perform the assignment of patients to caregivers according to a deterministic demand, Lanzarone and Matta [15] address the problem of assigning newly admitted patients by taking into account the stochasticity of new patient's demand. Recently considered, the uncertainty of the demand has also been studied by Shi et al. [19] who consider uncertain demand as a fuzzy variable to represent the quantity of drugs to deliver for each patient.

Besides, the concept of the penalty costs in the case of soft time window is subject to different representations depending on the researcher's choices. As an example, Bertels and Fahle [7] have modeled the cost function for the non-satisfaction of the soft constraints as a factor proportional to the earliness or the lateness of the vehicle to the patient's home. For each job, a penalty is applied if the vehicle arrives too early or too late. The sum of the penalties is then weighted in order to be added to the total travel time needed for the objective function. Also, Trautsamwieser and Hirsch [20] have studied the home health care problem with soft time windows on both nurses and clients. They used a metaheuristic based on Variable Neighborhood Search (VNS) to solve some randomly generated instances and real life instances provided by the Austrian Red Cross.

In this paper, the presence of the synchronization constraint between vehicles that is found frequently in the home health care application has to be pointed out. While most publications in the literature on the vehicle routing problem consider time window, the presence of synchronization constraints is uncommon. Synchronization constraints imply that some visits require multiple staff members at the same time. Therefore, the schedules of some staff members have to be synchronized (without mentioning previously which staff members are implied) at some point during the day. Eveborn et al. [11] consider these constraints in the routing and scheduling of a health care structure in Sweden. They developed a decision support system to aid the planners to quickly generate a valid and optimized schedule. In order to solve the synchronization constraints, the common technique is that the visits which require multiple staff members are handled by splitting those visits into two and then constraining the time those visits have to be performed. Similarly, Bredström and Rönnqvist [9] suggested a mathematical programming model for the vehicle routing and scheduling problem with time window and synchronization constraints as well as an optimization based heuristic. They tested the performance of their approaches on a set of data with a different number of customers, vehicles, synchronized visits and time window sizes. These data have later been also used by Afifi et al. [5] to test the performance of their simulated annealing based algorithm with a local search procedure. They showed that the local search procedure was efficiently adapted to handle the synchronization constraints. Classified by Drexl [23], different types of synchronization constraint do exist in the literature. Regarding the home health care problem, the synchronization constraint belongs to the type of operation synchronization, either as an operation synchronization with precedences or as an exact operation synchronization. These two types of synchronization have been considered by Kergosien et al. [14] in their integer linear program, in the branch-and-price algorithm of Rasmussen et al. [18] or in the adaptive variable neighborhood search algorithm of Mankowska et al. [17]. In this paper, only the exact operation synchronization is considered since the test instances used for the experiments do not include any kind of precedence between the realization of the cares.

All the previously cited works consider that the time window for performing the care is a hard time window. A hard time window for a job represents the time interval in which the job has to start. Thus, the vehicle has to arrive at the customer location absolutely within the time window without exceeding the time window limits. While a hard time window has to be met, a soft time window is seen as a preference which may be not fully satisfied. In this case, the objective function would then be overloaded by a penalty cost. Moreover, in existing papers, synchronization constraints are considered as hard constraints, i.e. the two vehicles have to be at the exact same time or without exceeding a maximal arrival time difference at the same place.

In this paper, we introduce soft constraints for time windows and synchronization. Indeed, a better solution in terms of total traveling time of the vehicles could be found by being more flexible on patients time windows. Moreover, the concept of soft time windows is extended to synchronization constraints. A penalty function whose value depends on the arrival time gap between both vehicles is suggested. Indeed, in the case of lifting a heavy patient, the two caregivers do not need to be together all of the time but only a few seconds to lift the patient. In practice, the cares provided by the two caregivers are mainly independent so that the first caregiver coming to the patient's home can start performing a part of their care until the second caregiver arrives at the patient's home.

The main contributions of this paper are the study of a variant of the home health care routing and scheduling problem with soft constraints for time windows and synchronization and the proposition of a memetic algorithm to solve this problem. Compared with the state of the art on, this variant brings more flexibility to the problem modelization. Moreover, the introduction of the soft constraints for time windows and synchronization to the home health care problem can be reduced to a problem with hard constraints when considering very high penalties for violating the soft constraints. This variant of the home health care routing and scheduling problem is original to the best of our knowledge.

In the home health care field, this variant allows the coordinating nurse to define more precisely the wishes of the patients to receive their care, in order to improve the quality of the planning obtained and the satisfaction of the patients. This accuracy is not possible in a modelization with only hard constraints.

Coordinating nurses will be able to indicate accurately which patients need to be treated during their time window of availability (e.g. because they have to work the rest of the time or if the care must be done at a specific time). Otherwise, patients with less time constraints during the day and mostly at home (e.g. elderly people) will be able to receive their care more easily outside the time period of availability originally planned.

On the same principle, synchronization of routes, in order to have two caregivers in the same patient's home at the same time, is variably flexible depending on the care to be performed. If the task to perform together is short (such as lifting or returning a patient), then the coordinating nurse will be able to give more flexibility to the simultaneous presence of the two caregivers, since they will need to be together only during a small part of the care. Then, the caregivers will be able to carry out their own care for the patient separately.

In addition, this paper provides a memetic algorithm featuring two original crossover operators in order to solve the problem. Experimental results highlight the efficiency of the proposed approach thanks to a comparison with best-known results from the literature as well as the commercial optimization solver Gurobi. As studied by Hiermann et al. [12], memetic algorithm outperforms other studied metaheuristic on a similar variant of the home health care problem.

This paper is outlined as follows. In the next section, characteristics of the studied system are detailed. In Section 3, the problem is formulated as a mixed integer programming model with a description of the various parameters, variables and constraints taken into account. In Section 4, a description of the memetic algorithm is provided. Section 5 unveils the numerical results of the experiments, a description of the test instances and an analysis of the results. Finally, some concluding remarks, as well as some perspectives on future works, will be drawn in the last section.

2. System definition

Home care services scheduling and routing problem for several types of staff member is defined as follows. Given a set of health care staff members and a set of visits to be performed at the patients' location, the goal is to find a valid schedule and a route on a one-day period for each staff member. The resulting planning has to indicate which visit has to be performed by which staff member and when the visit should start. Each staff member has a defined working time window which means they work only during this time period. Staff members have an associated health care office from which they have to start and finish their working day. Furthermore, the staff members have a fixed salary which does not depend on the actual amount of time they work during the day. Each staff member uses the same transportation mode (i.e. a car provided by the home health care company). Working overtime is not allowed due to an increased cost of work for the company. Home health care staff members are either nurses or auxiliary nurses. Each staff member has a different fixed salary which is based on skills and experience.

The time at which a staff member has to start a visit is restricted by a time window. Consequently, the patient and the home health care company agree on a time window in which the care has to start. However, this patient time window might be partially not satisfied by starting the visit shortly before or after the client preference. Each patient time window is specific for each visit depending on the patient availability or the type of care to provide. To that end, the level of tolerance to respect the time window is different depending on the care to be provided. The visits are carried out by only one category of staff member (e.g. an injection is performed only by a nurse), therefore some visit/staff member combinations are invalid. Some visits have to be synchronized when the health care has to be performed by two staff members simultaneously. In order to penalize the potential gap of arrival time between the two staff members, we consider a fixed soft time window of duration zero and the objective function is penalized when the arrival times of the staff members are different.

Finally, we assume that the main objective of the home health care company is to minimize the total time traveled by the staff members during their working day. Penalties will be added to the objective function if soft constraints are not satisfied.

3. Model description

In this part, the mixed integer programming formulation of the considered home health care problem is presented.

3.1. Core optimization problem

The home health care scheduling and routing problem is modeled on a graph $G = (N, A)$ where N is the set of nodes and A the set of arcs. The set of visits to perform is denoted O and the set of home health care offices is denoted P. Thus, $N = O \cup P$. Each visit is represented by a separated node in the graph, whether two or more visits are associated with the same physical location or not. For example, if a given client requires two visits during the day, a node will be created for each visit and they will both have the same geographical location. Using this information, the set of arcs is defined as $A = \{(i, j)|i, j \in N, i \neq j\}$. Each arc $(i, j) \in A$ has a distance d_{ij}.

The planning period is bounded by the interval $[l^T, u^T]$.

The set of caregivers is denoted S. For each caregiver $i \in S$, a working hard time window $[\alpha_i, \beta_i]$ is known. The set $R = \{n, u\}$ represents the possible job role that can have caregivers. Nurses have the job role n and the auxiliary nurses have the job role u. The association of a job role to a staff member i is defined by the binary parameter η_i. The parameter η_i is equal to n if the staff member i is a nurse and equal to u otherwise.

The home health care offices are located in different places and therefore have different locations. The binary parameter $\gamma_i^k = 1$ if the staff member i is associated with the home health care office k, and zero otherwise.

Moreover, for each visit $i \in O$ at the patient's home, a duration c_i represents the time needed to perform the care. A soft time window $[a_i, b_i]$ represents the availability of the patient to receive a care. The binary parameter δ_{ij} is used when two visits i and j are synchronized. $\delta_{ij} = 1$ if the visits i and j are synchronized, and 0 otherwise.

Finally, the staff member qualification needed to perform the job i is defined by ρ_i. If the visit requires a nurse, $\rho_i = n$, otherwise the visit requires an auxiliary nurse and $\rho_i = u$.

3.2. Soft time window and synchronization constraints

In this section, the concept of the penalties applied to the non-satisfaction of the soft time windows and synchronization constraints is detailed.

First, we define two zones that cause penalties when the soft constraints are not satisfied. If the violation of the soft constraint is within a reasonable gap, then some normal penalties will be added to the objective function. However, if most of the care is performed outside of the patient time window or if the shared visits are too far apart, then some extra penalties will be added.

Secondly, we suggest that the penalties applied to a care if the starting time of it is out of the patient time window is defined by a piecewise-linear function f_i. Piecewise-linear penalty function are used to apply penalties proportional to earliness or lateness of the arrival time. This concept of penalty function have been mainly applied to soft time window such as Bertels and Fahle [7] and Triki et al. [24].

To that end, the amount of the penalty applied to the objective function for the care i will depend on the starting time t_i of the care. The penalty function increase by a penalty cost per minute v_i^t in the normal zone. The penalty cost per minute is multiplied by an additional coefficient p^t if the starting time of the care i belongs to the extra zone. The time value which separates the normal and the extra zones is denoted τ_i for the patient i. It corresponds to a defined percentage λ of the visit duration. Consequently, we consider that $\tau_i = \lambda \times c_i$.

The Fig. 1 illustrates the value of the penalty function $f_i(t_i)$ depending on the arrival time of a staff member at the visit i.

Finally, we define the penalty function $f_i(t_i)$ as follows :

$$f_i(t_i) = \begin{cases} (\tau_i + (a_i - t_i - \tau_i) \times p^t) \times v_i^t, \ if \ t_i < a_i - \tau_i \\ (a_i - t_i) \times v_i^t, \ if \ t_i \in [a_i - \tau_i, a_i[\\ 0, \ if \ t_i \in [a_i, b_i] \\ (t_i - b_i) \times v_i^t, \ if \ t_i \in]b_i, b_i + \tau_i] \\ (\tau_i + (t_i - b_i - \tau_i) \times p^t) \times v_i^t, \ if \ t_i > b_i + \tau_i \end{cases} \tag{1}$$

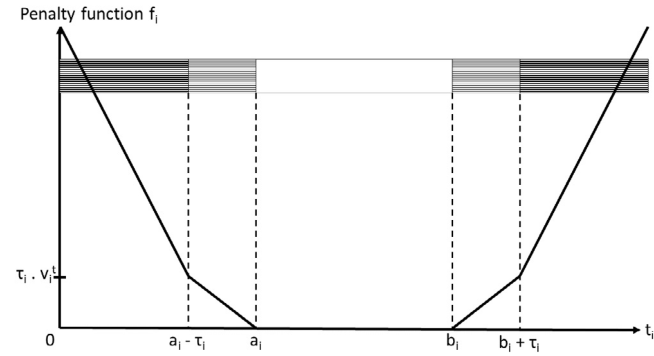

Fig. 1. Value of the penalty function f_i depending on the arrival time. Starting visit before a_i or after b_i is penalized for each minutes of non-satisfaction.

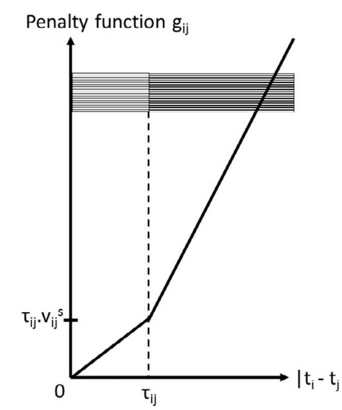

Fig. 2. Value of the penalty function g_{ij} depending on the arrival times of both caregivers.

Similarly, we suggest that the penalty applied to the synchronization of the visits i and j if the arrival times of both caregivers are different is defined by a piecewise-linear function g_{ij}. The time value which separates the normal and the extra zones is denoted τ_{ij} for the shared visits i and j. It corresponds to a defined percentage λ of the visit duration. Consequently, we consider that $\tau_{ij} = \lambda \times c_i$. The penalty function g_{ij} increases by a penalty cost per minute v_{ij}^s in the normal zone. The penalty cost per minute is multiplied by an additional coefficient p^s if the difference of arrival times is above τ_{ij} and belongs to the extra zone.

The Fig. 2 illustrates the value of the penalty function $g_{ij}(t_i, t_j)$ depending on the arrival time of the caregivers i and j.

Finally, we define the penalty function $g_{ij}(t_i, t_j)$ as follows :

$$g_{ij}(t_i, t_j) = \begin{cases} 0, \ if \ t_i = t_j \\ |t_i - t_j| \times v_{ij}^s, \ if \ |t_i - t_j| \leq \tau_{ij} \\ (\tau_{ij} + p^s \times (|t_i - t_j| - \tau_{ij})) \times v_{ij}^s \\ \quad , \ if \ |t_i - t_j| > \tau_{ij} \end{cases} \tag{2}$$

3.3. Problem formulation

In this part, the formulation of the problem is presented. The problem formulation is original to the best of our knowledge. It can however be reduced to the existing formulation with hard time window and synchronization constraints suggested by Bredström and Rönnqvist [9] by setting high penalties for violating the soft constraints.

The primary decision variable used in the mathematical model is :

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

The secondary decision variable used is :

t_i = Starting time of the visit i

Similarly, as the reality, the planning period considered is a single day.

The notation of the parameters for the home health care routing and scheduling problem are summarized below :

N: set of nodes
O: set of visits
S: set of staff members
P: set of home health care offices
R: set of job roles
$Sync$: set of synchronized visits
$[l^T, u^T]$: planning period of time
d_{ij}: distance between the nodes i and j
c_i: duration of the visit i
$[\alpha_i, \beta_i]$: working time window of the staff member i
$[a_i, b_i]$: availability time window of the patient i
η_i: qualification of the staff member i
ρ_i: qualification level required to perform the visit i
γ_i^k: association of the staff member i to the office k
δ_{ij}: indicates if i and j are synchronized visits
p^t: coefficient value for time windows non-satisfaction
p^s: coefficient value for synchronized visits non-satisfaction
λ: percentage used for the penalties computation
M: a large number

The mixed-integer programming modelization of the home health care routing and scheduling problem is formulated as follows :

$$min \sum_{i,j\in N} \sum_{k\in S} x_{ij}^k \times d_{ij} + \sum_{i\in O} f_i(t_i) + \sum_{i,j\in O} \delta_{ij} \times g_{ij}(t_i, t_j) \quad (3)$$

subject to :

$$\sum_{\substack{j\in N \\ i\neq j}} \sum_{k\in S} x_{ij}^k = 1 \qquad \forall i \in O \quad (4)$$

$$\sum_{j\in O} x_{ij}^k = \sum_{j\in O} x_{ji}^k = \gamma_k^i \qquad \forall i \in P, k \in S \quad (5)$$

$$\sum_{\substack{j\in N \\ i\neq j}} x_{ij}^k = \sum_{\substack{j\in N \\ i\neq j}} x_{ji}^k \qquad \forall i \in N, k \in S \quad (6)$$

$$t_j \geq t_i + c_i + d_{ij} + ((\sum_{k\in S} x_{ij}^k) - 1) \times M \qquad \forall i \in N, j \in O, i \neq j \quad (7)$$

$$t_i \geq \alpha_k + ((\sum_{\substack{j\in N \\ i\neq j}} x_{ij}^k) - 1) \times M \qquad \forall i \in O, k \in S \quad (8)$$

$$\beta_k \geq t_i + c_i + d_{ij} + (x_{ij}^k - 1) \times M \qquad \forall i \in O, j \in P, k \in S, i \neq j, \gamma_k^j = 1 \quad (9)$$

$$\sum_{\substack{j\in N \\ i\neq j}} x_{ij}^k = 0 \qquad \forall i \in O, k \in S, \eta_k \neq \rho_i \quad (10)$$

$$x_{ij}^k \in \{0, 1\} \qquad \forall i \in N, j \in N, k \in S$$
$$t_i \in T \qquad \forall i \in O \quad (11)$$

The objective function (3) minimizes the total traveling time of the staff members added to the penalties of the patients time window non-satisfaction and the synchronized visits non-respect. The penalties for early or late non-satisfaction of the patients time window or synchronized visits are split into two parts whether they incur a normal penalty or an extra penalty depending on the importance of the non-satisfaction.

Constraint (4) ensures that a visit is done by only one staff member. The departure and arrival of the staff members to their associated home care office is ensured by the constraint (5) while the flow conservation is guaranteed by the constraint (6).

The constraint (7) ensures that during their route, the staff members have enough time between two visits to perform the first one and then go to the location of the following patient. Moreover, the staff members work only during their own hard time window. They visit some patients during this period of time but they have to be back to their home health care office before the end of the time window according to the constraints (8) and (9).

Moreover, the constraint (10) aims to associate a staff member who has the requested qualification to a visit for performing the health care.

Finally, the constraint (11) defines the domain of the variables.

Note that the penalty functions defined by the Eqs. (1) and (2) are linearizable to obtain a MILP model.

The memetic algorithm we suggest to solve the home health care routing and scheduling problem is described in the next section.

4. Memetic algorithm

Among the growing research area of evolutionary computation, the choice has been made to implement a population-based approach like a memetic algorithm. Memetic algorithm was first introduced by Moscato [25] as the combination of evolutionary algorithms with local search procedure. The main benefit of the memetic algorithm is to combine the global search of evolutionary algorithm with the local search to improve individual solutions. According to Alba [26], "the balance between diversification and intensification is important, on one side to quickly identify regions in the search space with high-quality solutions, and on the other side not to waste too much time in regions of the search space which are either already explored or which do not provide high-quality solutions". For instance, diversity could be lost if an aggressive local search was applied to every individual of the population that would lead the algorithm to a local optimum without any possibility to escape the local optimum easily.

In this paper, the memetic algorithm is the hybridization of a genetic algorithm with a local search procedure. In order to give an overview of the structure of a memetic algorithm, Fig. 3 represents the flow chart of a memetic algorithm.

In the following parts of this section, the different phases of the memetic algorithm will be detailed.

4.1. Generation of the initial population

As the memetic algorithm is a population-based metaheuristic, the algorithm starts by the generation of an initial population. In order to generate the initial population, an insertion heuristic has been developed. An insertion heuristic is a really intuitive approach for constructing the solutions.

In order to give an overview of the insertion heuristic procedure, the following algorithm 1 details the different steps.

Algorithm 1 Insertion heuristic procedure

```
function INSERTIONHEURISTIC
    sol ← createStartingSolution();
    while isStoppingCriteriaReached ≠ true do
        result ← insertVisitWithRegret(sol, unRoutedNodes);
        if result = false then // insertion impossible
            localSearch(sol);
            result ← insertVisitWithRegret(sol, unRoutedNodes);
            if result = false then
                deteriorateSolution(sol);
            end if
        end if
        if getSize(unRoutedNodes) = 0 then
            localSearch(sol);
            updateBestSolutions(sol, bestSolutions);
            removeLongestRoutes(sol, unRoutedNodes);
            removeFurthestVisits(sol, unRoutedNodes);
            removeHighPenaltyVisits(sol, unRoutedNodes);
            removeRandomVisits(sol, unRoutedNodes);
        end if
    end while
    return bestSolutions;
end function
```

The basic idea is to start with a single visit as subtour and then insert the rest of the visits by some heuristic. In this case, the heuristic starts by creating as many tours as the needed number of routes (i.e. the number of staff members). These initial tours are just starting from the depot and returning to it. Then, one visit is added for each route. After that step, each route contains one visit to perform. The remaining part of the algorithm is executed as long as time remains.

At each iteration, an unrouted visit is added to one of the routes depending on the insertion cost and the feasibility of the insertion. In order to find the possible insertion positions, the regret value is computed for each unrouted visit. As explained by Potvin and Rousseau [27], the regret value of a visit is the difference of insertion cost between the first and second best insertion moves of this visit. The visit with the highest regret is inserted first.

In the case no insertion is possible but unrouted visits remain, a local search iteration is performed on the current incomplete solution and the algorithm tries again to find a possible insertion move. If no possible insertion move is found, a route is randomly cleared, and all the visits contained in this route are moved in the list of unrouted nodes. The empty route is then filled again with one of the unrouted visits randomly chosen.

When the solution is complete and valid, the local search procedure is processed as long as an improvement is found by any of the operators. The local search procedure used is the same as the one used by the memetic algorithm described in Section 4.5.

In order to generate new solutions from previous ones, we propose to use several destruction operators and then insert the unrouted visits thanks to the insertion procedure.

The destruction operators are the followings: removal of the routes with the highest mean objective value, removal of the visits which incurs a high increase of the traveling time, removal of the visits incurring high penalties and finally random removal of some visits.

All these previous operations allow us to restart the process of creating a solution with a newly incomplete solution based on the good parts of the previously generated one for the intensification of the algorithm and on some random actions for the diversification of the next valid solution.

The initial population is handled by the insertion heuristic. The newly improved solution is added to the population if the solution is unique (i.e. different from any of the solutions already contained in the initial population) and if its fitness is better than the worst solution of the population. In that case, the worst solution in terms of fitness is removed from the population and the new solution is inserted instead.

At the end of the time limit, the memetic algorithm keeps the best required number of solutions which are all unique to establish its initial population.

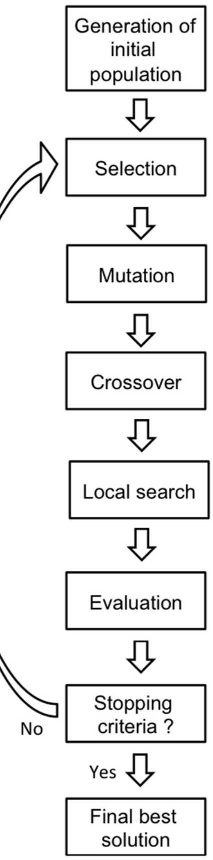

Fig. 3. Memetic algorithm structure — Hybridization of a genetic algorithm and a local search operator.

4.2. Selection procedure

Selection is the stage of the memetic algorithm where individuals are selected from a population in order to breed a new generation. In the selection procedure, the individuals are selected through a fitness-based process where the individuals with the lowest objective value are more likely to be selected. Indeed, the objective value of each individual of the population has to be known in order to process this stage.

There are different techniques of selection to pick the individuals from the population. In the selection process, a combination of two selection methods has been chosen.

First, a fitness proportionate selection will choose a percentage θ_f of the individuals of the next population. Fitness proportionate selection works as follows. Each individual has a fitness level which is associated with a probability of being selected. In this case, the lower the fitness is and the greater the probability of being selected is.

To do so, the fitness value Ω_i of an individual i is equal to the inverse of the value of its objective function. Then, its probability of being selected is equal to $\frac{\Omega_i}{\sum_{j=1}^{N}\Omega_j}$ where N is the number of individuals in the population.

Secondly, the remaining θ_r of the population are determined by rank based selection. Rank based selection tends to avoid premature convergence by reducing selection pressure which occurs in early generations when large fitness differences between individuals are frequent. Similarly to the fitness proportionate selection, the probability of being selected is computed in the same way. However, the fitness value of an individual corresponds to their ranking, the worst solution receiving a rank of 1 and the best one a rank of N.

It should be remembered that the members of the next generation are not selected yet, the selected individuals will go through the crossover process. An individual may be selected as a parent multiple times for the next step of the algorithm.

4.3. *Crossover operators*

Crossover operators are used for creating new solutions from population's genetic information analogously to reproduction. Crossover facilitates inheritance of characteristics by an offspring from its parents.

For each crossover operator, a dynamic probability that the operator will occur on the selected individuals is proposed. Indeed after each iteration, the probabilities of the crossover operators are adjusted. The probability of a crossover operator depends on the quality of its generated children solution. Therefore, the more the individuals generated by a crossover operator are good and the higher the probability of occurring will be. The crossover operators probabilities range from 75% to 95%. These two bounds have been set in order to keep an overall high probability of crossover for each operator since this configuration provides the best results according to our experiments.

There are many ways to implement and reproduce the crossover process. The operators used in the crossover procedure are detailed below, including the synchronization breakpoint crossover and the route mix crossover operators which are a new contribution to the extent of our knowledge.

4.3.1. *Synchronization breakpoint crossover (see Fig. 4)*

A synchronized pair of visits (x, y) on both parents organism strings is selected. Since synchronized visits consist of actually two visits with a desired similar arrival time, the following process is repeated for each visit of the pair. A one-point crossover is processed with the synchronized visits as a crossover point. The crossover is applied not on the full parent organism but only on the routes which contain the crossover point. The routes which do not contain the synchronized visits are not impacted. If some visits become unrouted in the resulting children, they are added following their least cost insertion.

4.3.2. *Route mix crossover (see Fig. 5)*

As long as the child is not valid, the routes of both parents are gathered and sorted by ascending fitness. The first route of the list (i.e. the best one) is added to the child's routes. The visits contained in the added route are removed from the set of remaining routes to add in order to avoid adding twice the same visit. If some visits become unrouted when the required number of routes is reached, they are added following their least cost insertion.

4.3.3. *One-point crossover*

A single visit on both parents representation is selected. All data beyond that point in either organism is swapped between the two parents organism. The resulting organisms are the children. Cutting of the parent's routes is kept up to the crossover point. The remaining part after the crossover point is cut randomly for the children to obtain the desired number of routes

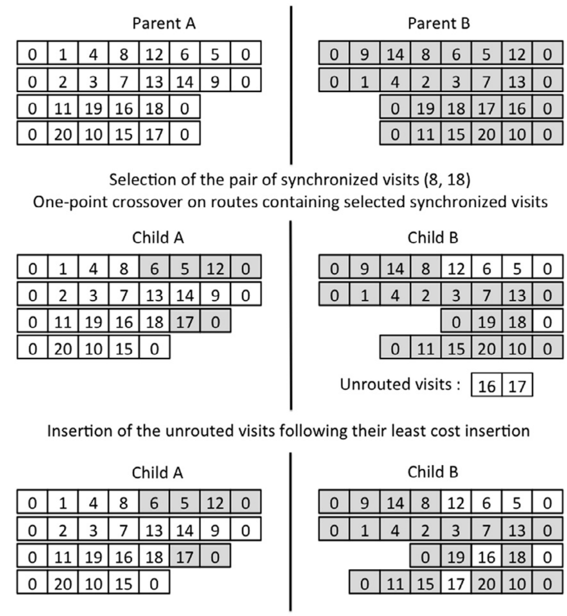

Fig. 4. Illustration of the synchronization breakpoint crossover process.

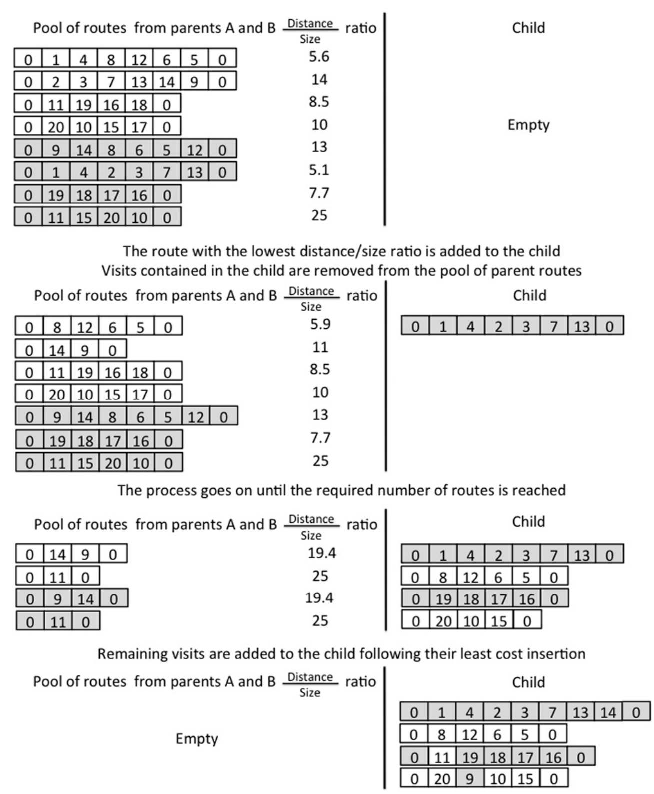

Fig. 5. Illustration of the route mix crossover process.

4.3.4. *Two-point crossover*

In a similar way to the one-point crossover, two visits on both parents representation are selected. Everything between the two points is swapped between the parent's organism while everything out of the two points in the children organism is cut randomly to obtain the desired number of routes.

4.3.5. One path exchange crossover

Initially in this operator, the children are the copy of their parents. However, one randomly chosen path of the parent of which they are not the copy is conserved and copied in the child solution.

The crossover procedure is not systematically applied on a pair of parent solutions but depends on a crossover probability. Each probability to apply a crossover operator is unique. If needed, the probabilities are updated at every iteration in order to always give to the operator creating the children with the best average fitness the highest probability.

4.4. Mutation operators

The main role of the mutation operators is to preserve the genetic diversity of the population. Mutation operators change significantly a solution from its initial state and greatly help the memetic algorithm to avoid local optima.

For each selected individual, the probability of applying a mutation operator has been set to 10%. This value has been set in order to keep an overall low probability of mutation for each operator since this is the best configuration according to our experiments. In the case an individual is selected for the application of a mutation operator, the mutation operator applied is chosen randomly among the following mutation operators. The probabilities of being selected are equal between all mutation operators.

In the node exchange mutation operator, a randomly selected visit is exchanged with another randomly selected visit. The node relocation operator aims to select randomly a visit and to relocate the visit in another route of the solution. While the previous two operators focus on the routing part, the arrival time shift operator focus on the scheduling. In each route of the solution, a visit is selected randomly and the beginning time of the visit is updated with a change comprised in a certain interval of time.

After all the changes brought by the crossover and the mutation operators, the individuals of the population have to be evaluated before moving on to the next phase.

Moreover, some of the generated solutions might have become unfeasible. If so, a correction procedure is applied to get a feasible solution.

Finally, the improvement procedure of the algorithm is performed by a local search on separate individuals. The local search procedure is detailed in following section.

4.5. Local search procedure

Local search algorithms apply local changes iteratively to the solution until the stopping criterion is reached (local optima found, time bound elapsed, ...).

Since local search procedure is really time-consuming, a strategy has to be adopted to decide how often or on which solutions individual learning should be applied. In this case, the local search is applied only to a part of the population with the best fitness.

In order to give an overview of the local search procedure, the following algorithm 2 details the different steps.

The following local search operators have been adapted to the specificity of the problem considering soft constraints for time windows and synchronizations. Each operator returns a boolean indicating if an improvement has been made or not on the studied solution.

2-opt operator explores the possibilities of exchanging two paths that cross over each other in the same route. This classical local search operator is really effective for vehicle routing problems except when time windows are considered. Indeed, it is critical to preserve the route orientation when performing an exchange. Consequently, the same operator combines the 2-opt procedure with 2-opt* (initially introduced by Potvin et al. [28]). 2-opt* works similarly as 2-opt except that the exchange is performed on two different routes. Combining these two operators is particularly effective for this case since it is hardly possible for the 2-opt operator to find an improvement because of the preserved order of visits from the time windows.

Algorithm 2 Local search procedure

```
procedure LOCALSEARCH(Population p)
    sortPopulationByAscendingFitness(p)
    for i ← 1, getSize(p)/4 do
        improvement ← true
        while improvement = true do
            improvement ← twoOpt(p[i])
            if improvement = true then
                improvement ← twoOptStar(p[i])
            else
                twoOptStar(p[i])
            end if
            if improvement = true then
                improvement ← nodeExchange(p[i])
            else
                nodeExchange(p[i])
            end if
            if improvement = true then
                improvement ← nodeRelocation(p[i])
            else
                nodeRelocation(p[i])
            end if
        end while
    end for
end procedure
```

Moreover, the node exchange operator exchanges two visits with each other if the fitness of the solution is improved. All the different possibilities of exchange are studied. The first feasible move is realized and the procedure keeps going until no improvements are possible.

Finally to the node exchange operator, the node relocation operator consists of moving every routed visit from its current position to another one if the overall fitness of the solution is improved. Unlike the node exchange operator, only one visit is considered at a time. The first feasible move is realized and the procedure keeps going until no improvements are possible.

5. Numerical experiments

In this section, the test instances and the settings applied to the memetic algorithm (MA) are described. Then, the results obtained on the literature instances are compared with the best-known results or with an optimization solver considering hard and soft time window and synchronization constraints.

Indeed, the goal of comparing the memetic algorithm with hard and soft constraints is to evaluate the quality of the proposed algorithm on diverse configurations.

5.1. Test instances

The main goal of this part is to evaluate the efficiency of the proposed memetic algorithm for the home health care routing and scheduling problem. Moreover, the proposed algorithm is able to handle both hard and soft time window and synchronization constraints.

Firstly, the results obtained by the memetic algorithm are compared with the existing literature on the standard instances of

Table 2
Bredström and Rönnqvist test instances. The columns are : the number of visits |O|, the number of routes |S|, the number of synchronized visits |Sync|, the average distance between the visits AvD and the last five columns are the average time window size for each category.

Instance	|O|	|S|	|Sync|	AvD (h)	F(h)	S(h)	M(h)	L(h)	A(h)
1	20	4	2	0.3	0	1.5	2.1	2.9	9
2	20	4	2	0.27	0	1.7	2.2	3.0	9
3	20	4	2	0.25	0	1.5	2.4	3.0	9
4	20	4	2	0.39	0	1.8	2.9	3.9	9
5	20	4	2	0.27	0	1.3	2.1	3.2	9
6	50	10	5	0.33	0	1.4	2.3	3.1	9
7	50	10	5	0.31	0	1.6	2.5	3.4	9
8	50	10	5	0.32	0	1.5	2.4	3.2	9
9	80	16	8	0.29	0	1.5	2.3	2.9	9
10	80	16	8	0.23	0	1.6	2.6	3.6	9

Table 3
OptimHaD test instances. The columns are : the number of cares to be performed by a nurse $|O^{Nurse}|$ or by an auxiliary nurse $|O^{AuxNurse}|$, the number of available nurses $|S^{Nurse}|$ and auxiliary nurses $|S^{AuxNurse}|$, the number of home health care offices |P|, the number of synchronized visits |Sync| and the average distance between the visits AvD.

Instance	$\|O^{Nurse}\|$	$\|O^{AuxNurse}\|$	$\|S^{Nurse}\|$	$\|S^{AuxNurse}\|$	|P|	|Sync|	AvD (m)
A	103	6	6	1	1	2	4.28
B	40	0	2	0	1	0	2.82
C	40	0	2	0	1	0	2.90
D	23	6	1	1	1	1	2.04
BD	63	6	3	1	2	1	3.65
CD	63	6	3	1	2	1	4.16
BCD	103	6	5	1	3	1	5.38

Bredström and Rönnqvist [9] in the case of hard time window and synchronization constraints.

The instances are grouped into three categories depending on the number of clients. The first category contains five small-size instances of 20 clients, the second one contains three mid-size instances of 50 clients and the last category contains two instances of 80 clients.

For all instances, one day is considered to be nine hours long. Therefore, all the data related to the time in the instances have to be normalized on a daily nine-hour scale.

Moreover, in each instance, 10% of the visits are synchronized visits.

Finally, in order to cover the different time window possibilities, five groups of time window restrictions have been created respectively fixed time window (F), small time window (S), medium time window (M), large time window (L) and no time window restrictions (A).

For instance, in the tables presenting the results of the algorithms, the instance "3S" represents the instance 3 with small patients time window. Similarly, the instance "8A" represents the instance 8 with no time window restrictions.

As a summary, the characteristics of the test instances are presented in Table 2.

Secondly, the efficiency of the memetic algorithm is compared to an optimization solver in the case of soft constraints for time windows and synchronizations since there is no existing work on these instances to provide a fair comparison. Consequently, the memetic algorithm is evaluated on the same standard literature instances of Bredström and Rönnqvist [9] such as previously as well as on a novel set of instances named OptimHaD (see Table 3) based on the real-life data provided by a home health care company in France. Original data have been anonymized and slightly altered in order to introduce few synchronizations between visits.

The proposed instances[1] contain either 1, 2 or 3 home health care offices. Indeed, the instances A, B, C, and D correspond to a specific geographical area, while the instances BD, CD, and BCD are combinations of the four previous instances. As an example, the instance CD gathers in one instance the data of the instances C and D. Consequently, the instance CD includes 2 home health care office with a pre-assignment of the caregivers to their associated home health care office.

Time window restrictions are different for each visit depending on the type of care to be provided.

5.2. Experiments settings

In order to test the MA on the instances presented previously, the parameters that will update the behavior of the algorithm are defined.

One of the main settings of the algorithm to define is the computational time allowed for each instance. The computational time set for the insertion heuristic used to generate the initial population is defined to 2 min, for all instance sizes. Then, the computation time for Bredström and Rönnqvist [9] instances with 20 visits is set to 2 min and the computation time for instances with 50 and 80 visits is set to 10 min for the MA. In order to obtain a fair comparison with the best results obtained, we place ourselves in the same frame as the original authors by reusing the same calculation times than Bredström and Rönnqvist [9]. These values are the same as used by Bredström and Rönnqvist [9] in order to establish an accurate comparison of the results obtained. However, it should be pointed out to the reader that our solving method may take advantage of the progression of computer performance compared to experiments performed by the original authors.

Moreover, the computation time for OptimHaD instances is set to 2 min for the instances A, B and C (below 50 visits) and 10 min otherwise.

Regarding the insertion heuristic, the value of θ_1 has been set empirically to 0.9 in order to remove a high number of visits which deteriorate the quality of the solution. This value has been chosen since it removes an important part of the visits focusing on the visits occurring long detours. The parameter θ_2 has been set empirically to 80% in order to avoid reconstructing the same solution at the next iteration. A high value has been chosen in order to get an important focus on the diversity of the solutions generated by the insertion heuristic. Indeed, the solutions found by the insertion heuristic will be used as the initial population of the MA. Therefore, the initial population must be diverse to be able to browse the entire range of possible solutions.

Moreover, the values of θ_f and θ_r have been respectively set to 80% and 20% to provide a good balance between diversity and intensity of the selection procedure according to our experiments. Indeed, a focus is given on elitism since the computational time is limited but the diversity is maintained thanks to the rank-based selection.

After realizing several experiments, the value of the limit separating the two zones of penalty denoted τ_i has been set to $0.1 \times c_i$. Similarly, the time limit separating the two zones of penalty in the case of synchronized visits denoted τ_{ij} for the visits i and j is set to $0.1 \times c_i$ assuming $c_i = c_j$ in the experiments. The aim is to strongly penalize the large violations of the soft constraints. This value has been set empirically since the experiments realized with approaching values were providing similar results.

The planning period considered is a single day. Therefore, the period of time $[l^T, u^T]$ is set to the interval [0,1440].

[1] OptimHaD instances will be published at www.researchgate.net/profile/Jeremy_Decerle.

Table 4
Comparison of the results of the memetic algorithm with the commercial optimization solver Gurobi for soft time window and synchronization constraints - Bredström and Rönnqvist [9] benchmark.

Instance		Objective function (h)			Deviation ratio (%)	TW respect (%)		Average TW satisfaction (%)		Average sync. satisfaction (%)	
		MIP	MA	$\overline{\text{MA}}$		MIP	MA	MIP	MA	MIP	MA
1	F	4.67*	4.79	4.81	2.55	75	85	94.94	97.05	100	100
2	F	4.93*	4.95	4.96	0.34	95	95	97.06	97.83	100	100
3	F	4.82	4.90	4.93	1.62	70	80	97.56	99.11	99.14	100
4	F	7.10	7.18	7.18	1.11	70	85	98.92	98.97	100	99.43
5	F	5.07	5.32	5.37	4.72	45	90	92.29	96.96	99.79	94.93
6	F	38.93	13.10	13.12	−197.22	34	86	54.68	94.70	59.31	100
7	F	–	11.91	11.91	–	–	84	–	94.02	–	97.58
8	F	–	18.04	18.73	–	–	58	–	88.67	–	96.44
9	F	–	21.48	21.85	–	–	22	–	89.47	–	95.55
10	F	–	11.34	11.45	–	–	95	–	99.55	–	100
1	S	3.45*	3.55	3.55	2.86	75	100	96.47	100	100	100
2	S	3.85*	3.94	3.94	2.23	50	75	94.58	92.08	100	100
3	S	3.52	3.56	3.59	0.99	70	100	93.69	100	95	93.38
4	S	5.90	5.77	5.77	−2.22	90	95	97.53	97.71	79.41	100
5	S	3.64*	3.70	3.70	1.49	45	90	92.23	97.64	100	100
6	S	10.13	8.03	8.09	−26.06	68	90	86.55	99.33	76	100
7	S	–	7.91	8.02	–	–	96	–	99.02	–	100
8	S	–	9.02	9.11	–	–	96	–	99.20	–	100
9	S	–	11.63	11.76	–	–	95	–	98.80	–	100
10	S	–	8.80	8.84	–	–	100	–	100	–	100
1	M	3.44*	3.48	3.51	1.36	85	95	94.71	99.48	100	100
2	M	3.44*	3.45	3.47	0.24	90	95	98.80	99.30	90	100
3	M	3.31	3.33	3.33	0.61	90	100	99.39	100	100	100
4	M	5.42	5.36	5.36	−1.02	65	90	90.81	96.92	45	76.13
5	M	3.44*	3.49	3.49	1.34	75	85	90.21	90.41	100	100
6	M	–	7.59	7.71	–	–	90	–	98.52	–	100
7	M	–	7.25	7.39	–	–	90	–	99.05	–	100
8	M	–	8.41	8.46	–	–	92	–	98.90	–	100
9	M	–	10.90	11.05	–	–	95	–	98.99	–	100
10	M	–	7.97	8.12	–	–	98	–	99.89	–	100
1	L	3.32*	3.32	3.34	0.10	95	95	99.33	99.48	100	100
2	L	3.29*	3.33	3.36	1.06	90	95	99.22	98.09	50	100
3	L	3.27	3.28	3.29	0.34	90	95	99.32	99.87	100.00	100
4	L	4.97	5.11	5.12	2.87	85	95	97.98	99.03	0	100
5	L	3.27	3.31	3.31	1.04	80	90	92.98	99.28	100	100
6	L	–	7.32	7.35	–	–	92	–	96.75	–	100
7	L	–	6.83	6.97	–	–	96	–	98.72	–	100
8	L	–	7.95	8.02	–	–	92	–	98.53	–	100
9	L	–	10.34	10.45	–	–	91	–	98.75	–	100
10	L	–	7.80	7.83	–	–	97	–	99.48	–	100
1	A	3.04	3.04	3.04	0	100	100	100	100	0	0
2	A	2.85*	2.85	2.87	0	100	100	100	100	0	0
3	A	2.83*	2.84	2.85	0.31	100	100	100	100	61.04	67.74
4	A	4.21	4.17	4.23	−1.03	100	100	100	100	0	0
5	A	2.83*	2.85	2.89	0.51	100	100	100	100	50	40.62
6	A	–	6.44	6.51	–	–	100	–	100	–	100
7	A	6.82	5.92	5.95	−15.18	100	100	100	100	20	100
8	A	–	6.49	6.51	–	–	100	–	100	–	23.36
9	A	–	8.97	9.05	–	–	100	–	100	–	100
10	A	–	6.84	6.85	–	–	100	–	100	–	64.20

The computation of the penalty value for soft patients time window non-satisfaction v_i^t depends on the average distance between all patient locations AvD and on the time window length of the visit. If the time window of the patients is small, they have some important impediments that prevent them from being available for a longer time. Consequently, the more the patient time window is small, the greater the penalty value is.

In addition, the more the average distance between patient locations is important and the greater the value of the penalty will be. Since a high value of AvD tends to induce a high value of the total travel time needed by the staff members (that is to say the objective function), the value of the penalty has to be proportional to the total travel time of the staff members in order to have a roughly similar impact on the penalization of the objective function.

Consequently, the penalty value for each minute of non-satisfaction of the patients soft time window v_i^t for the visit i is defined as follows :

$$v_i^t = \begin{cases} \dfrac{AvD}{2 \times (b_i - a_i)} & \text{if } 2 \times (b_i - a_i) \geq 1) \\ AvD & \text{otherwise} \end{cases}$$

In the case of hard time window and synchronization constraints, the experiments have been run with a high penalty value so as to avoid any non-satisfaction of the soft constraints. To that end, the parameters v_i^t and v_{ij}^s have been set to a high value defined to 10 000.

In Section 3 of this paper, the formulation of the model has been detailed and several variables such as the penalty factor or the penalty value have been introduced.

Table 5
Computational time in seconds of Gurobi for soft time window and synchronization constraints - Bredström and Rönnqvist [9] benchmark.

Instance	F	S	M	L	A
1	207	201.19	21.85	1275.16	3600
2	1560.95	1126.82	479.43	554.73	3.09
3	3600	3600	3600	3600	177.59
4	3600	3600	3600	3600	3600
5	3600	1360.12	1127.53	3600	976.15
6	3600	3600	3600	3600	3600
7	3600	3600	3600	3600	3600
8	3600	3600	3600	3600	3600
9	3600	3600	3600	3600	3600
10	3600	3600	3600	3600	3600

The penalty factor for soft time windows p^t is set to the same value than the penalty factor for synchronized visits p^s. Consequently in the experiments, $p^t = p^s = 2$.

Similarly, the same penalty value (i.e. penalty applied for each minute of non-satisfaction of the soft constraint) is used for the soft time windows and the synchronization constraints non-satisfaction. As a result, for a job i, $v_i^t = v_{ij}^s$.

Also, the size of the population has been set empirically to 20 individuals because it shows a good balance during the experiments between the solution accuracy and the computational time needed to perform one iteration of the memetic and genetic algorithm. The use of larger populations would increase the needed computational resources to perform one iteration.

The experiments are performed on a computer using an Intel Xeon(R) CPU E5-1603 (@ 2.80 GHz) CPU and 8 GB of RAM memory.

Finally, the optimization solver Gurobi Optimization [29] (version 6.0.5) is used with a time limit of 60 min to perform a comparison of the performance in front of the memetic algorithm on the health care problem at home with soft time window and synchronization constraints.

The results presented in the next section are the best ones obtained among 10 runs on each instance.

5.3. Computational results

With the parameters defined previously, the memetic algorithm is first compared to an optimization solver with soft time window and synchronization constraints since there are no existing results on the test instances with soft constraints. This comparison aims to evaluate the algorithm proposed with the exact model, in order to assess its performance.

Then, the memetic algorithm is evaluated with hard time window and synchronization constraints in order to provide a fair comparison of its performance with existing work.

5.3.1. Soft time window and synchronization constraints

In this part, the memetic algorithm is evaluated against the optimization solver Gurobi on the home health care problem with soft time window and synchronization constraints. A star mark (*) is used in this column to indicate that the solution has been proven to be optimal by the solver.

In order to obtain a fair comparison between the memetic algorithm and the optimization solver Gurobi, the computational time of Gurobi is detailed in Tables 5 and 7.

In each column, the best solution obtained by the memetic algorithm out of the ten runs is presented in the column "MA" and the result obtained by the optimization solver is given by the column "MIP". As well, the average result obtained by the memetic algorithm of the ten test runs is presented in the column $\overline{MA}$.

The "Objective function" column represents the sum of traveling time and penalties occurred by the non-satisfaction of the soft constraints.

The column "Deviation ratio (%)" indicates the gap obtained by the memetic algorithm compared to the optimization solver. The lower the deviation ratio is and the better the result is. Consequently, the deviation ratio is computed as follows :

$$Deviation\ ratio = \frac{MA - MIP}{MIP}$$

Moreover, the column "TW respect (%)" indicates the percentage of visits time window that are respected (i.e. when the staff member starts the care within the patient time window).

The column "Average TW satisfaction (%)" represents for all visits the average percentage of time of the care duration which is performed inside of the patient time window, denoted by the variable $AvTW$. It aims to quantify the respect of the soft patients time window. The amount of time of the care i which is out of the patient time window is computed by the variable Δ_i. Consequently, the value of $AvTW$ is computed as follows:

$$AvTW = \frac{\sum_{i \in O} \frac{c_i - \Delta_i}{c_i}}{|O|}$$

Finally, the last column "Average sync. satisfaction (%)" represents the average percentage of time of the synchronized visits duration where both staff members are present, represented by the variable AvS. The goal is to assess the respect of the synchronization constraints. The number of minutes of difference between the arrival times of the synchronized visits i and j is defined by the variable Δ_{ij}. Consequently, the value of AvS is computed as follows:

$$AvS = \frac{\sum_{(i,j) \in Sync} \frac{c_i - \Delta_{ij}}{c_i}}{|Sync|}$$

These different criteria will be used to compare the quality of the solutions.

From the benchmark of Bredström and Rönnqvist [9], the memetic algorithm performs very well (see Table 4) since it maintains an average gap of 1%–2% with the optimization solver when the latter finds a feasible solution. In addition, among the 28 instances where the optimization solver finds a feasible solution within its time limit, the memetic algorithm performs much better in 6 instances. Moreover, the memetic algorithm finds a feasible

Table 6
Comparison of the results of the memetic algorithm with the commercial optimization solver Gurobi for soft time window and synchronization constraints — OptimHaD benchmark.

Instance	Objective function (h)			Deviation ratio (%)	TW respect (%)		Average TW satisfaction (%)		Average sync. satisfaction (%)	
	MIP	MA	$\overline{MA}$		MIP	MA	MIP	MA	MIP	MA
A	171.76	106.60	107.65	−37.94	83.49	100	85.63	100	50	100
B	36.07	36.18	36.28	0.30	92.5	100	92.58	100	–	–
C	39.06	39.06	39.31	0	100	100	100	100	–	–
D	31.97*	32.24	32.44	0.84	93.10	100	96.21	100	100	100
BD	71.25	70.11	70.36	−1.60	100	100	100	100	100	100
CD	90.54	71.43	71.99	−21.11	92.75	100	96.18	100	100	100
BCD	206.69	109.10	110.15	−47.22	88.07	100	88.07	100	88.07	100

Table 7
Computational time in seconds of Gurobi for soft time window and synchronization constraints — OptimHaD benchmark.

Instance	Computational time
A	3600
B	3600
C	3600
D	502.28
BD	3600
CD	3600
BCD	3600

solution for each of the benchmark instances. As well, the results highlight the efficiency of the memetic algorithm which maintains a high satisfaction of the soft time window and synchronization constraints.

Concerning the OptimHaD benchmark, the memetic algorithm shows outstanding results (see Table 6) compared to the optimization solver. As a matter of fact, the MA outperforms Gurobi on large instances and provides equivalent performance on smaller instances. Indeed, the results of the MA are much better on 4 of the 7 instances in comparison with the optimization solver. Similarly, as the previous benchmark instances, the respect of the soft constraint is really high since all the patients time window and synchronized visits are satisfied by the memetic algorithm.

Overall, the computational results highlight the efficiency of the memetic algorithm. The proposed algorithm provides great results either with hard or soft time window and synchronization constraints comparing to the best-known results or to the commercial optimization solver Gurobi. In addition, the memetic algorithm has been evaluated on instances of up to 103 patients. Since more than 93% of the home health care structures can support less than 100 patients (1953 out of 2095 structures in France in 2008 according to Bertrand [3], the memetic algorithm could then be applied to a majority of home health care offices in the French territory in this scale of problem size.

5.3.2. Hard time window and synchronization constraints

In order to provide a fair comparison, the memetic algorithm has been tested on the benchmark suggested by Bredström and Rönnqvist [9] with hard time window and synchronization constraints. The objective is to minimize the total traveling time of the caregivers. To do so, high penalty values are set in order to avoid any violation of the soft constraints.

The total traveling time of the staff members computed by the memetic algorithm is compared with the best results of the literature with hard time window and synchronization constraints shown in the column "Best known". The results provided in this column have been found by exact and heuristic methods. A star mark (*) is used in this column to indicate that the solution has been proven to be optimal by an exact method for the hard constraints problem.

The best known results are proposed by Afifi et al. [5], Bredström and Rönnqvist [9] and Gayraud [30]. Computational results are presented in the Table 8.

Note that the instances 6F, 7F, 8F, 9F, 10F and 6A, 7A, 8A, 9A, 10A do not have any best-known value because no results have been provided in the literature for the resolution of these instances with hard time window and synchronization constraints. To that end, the results of the algorithms on these instances are provided for future considerations but cannot be used for comparison.

The results highlight the efficiency of the memetic algorithm. Indeed, the memetic algorithm reaches the best known solution for all the small size instances. As well, the algorithm maintains a small gap (around 1–2%) with the best-known solution on the mid-size instances and around 4–5% for the large-size instances. In addition, the optimal solutions have been found for the instances 6F and 7F by the memetic algorithm, which has been proven to be optimal thanks to an optimization solver.

Moreover, we found out that among the original instance set, 3 instances (8F, 9F, and 10F) do not have any feasible solution, by trying to solve the instances using the optimization solver Gurobi.

Overall, the memetic algorithm presented in this paper – designed for handling both hard and soft time window and synchronization constraints – shows great efficiency compared to the existing results from the literature by reaching the best known solution for most of the instances. In addition, the MA maintains a low gap compared to the best-known results for larger instances.

Table 8
Comparison of the results of the memetic algorithm with the best known results for hard time window and synchronization constraints - Bredström and Rönnqvist [9] benchmark.

Instance		Traveling time (h)			Deviation ratio (%)
		Best known	MA	$\overline{\text{MA}}$	
1	F	5.13*	5.13	5.13	0
2	F	4.98*	4.98	4.98	0
3	F	5.19*	5.19	5.19	0
4	F	7.21*	7.21	7.21	0
5	F	5.37*	5.37	5.37	0
6	F	–	14.45*	14.45	–
7	F	–	13.02*	13.03	–
8	F	–	Infeasible	Infeasible	–
9	F	–	Infeasible	Infeasible	–
10	F	–	Infeasible	Infeasible	–
1	S	3.55*	3.55	3.55	0
2	S	4.27*	4.27	4.27	0
3	S	3.63*	3.63	3.63	0
4	S	6.14*	6.14	6.14	0
5	S	3.93*	3.93	3.93	0
6	S	8.14*	8.28	8.31	1.72
7	S	8.39*	8.41	8.46	0.24
8	S	9.54*	9.54	9.59	0
9	S	11.93	12.48	12.59	4.61
10	S	8.6	8.80	8.97	2.33
1	M	3.55*	3.55	3.55	0
2	M	3.58*	3.58	3.58	0
3	M	3.33*	3.33	3.33	0
4	M	5.67*	5.67	5.67	0
5	M	3.53*	3.53	3.53	0
6	M	7.7	7.90	8.02	2.60
7	M	7.48	7.53	7.65	0.67
8	M	8.54*	8.76	8.89	2.58
9	M	10.92	11.49	11.86	5.22
10	M	7.62	8.08	8.23	6.04
1	L	3.39*	3.39	3.39	0
2	L	3.42*	3.42	3.42	0
3	L	3.29*	3.29	3.29	0
4	L	5.13*	5.13	5.20	0
5	L	3.34*	3.34	3.34	0
6	L	7.14*	7.39	7.56	3.50
7	L	6.88	6.98	7.04	1.45
8	L	8.14*	8.23	8.33	1.11
9	L	10.49	10.96	11.11	4.48
10	L	7.75	7.99	8.17	3.10
1	A	3.12	3.12	3.12	0
2	A	2.97	2.97	2.97	0
3	A	2.85	2.85	2.93	0
4	A	4.29	4.29	4.29	0
5	A	2.92	2.92	2.95	0
6	A	–	6.48	6.66	–
7	A	–	5.92	6.07	–
8	A	–	6.74	6.95	–
9	A	–	9.18	9.51	–
10	A	–	7.15	7.29	–

6. Conclusion

In this publication, we have introduced a new variant of the home health care routing and scheduling problem with soft time windows and synchronization constraints. This variant brings more flexibility to the problem by allowing the coordinating nurses to indicate accurately which patients need to be treated during their time window of availability. As well, the synchronization of routes may be adjusted depending on the care to be performed. We developed a memetic algorithm featuring two original crossover operators to the extent of our knowledge. In order to evaluate the quality of the proposed algorithm, experiments have been performed on benchmark instances from the literature as well as a new instance set based on real life data. The algorithm has been compared to the best-known result when considering hard time window and synchronization constraints in order to provide a fair comparison. Similarly, experiments have been run with soft time window and synchronization constraints to compare the performance of the memetic algorithm with the commercial optimization solver Gurobi.

The results clearly show the efficiency of our memetic algorithm to minimize the total traveling time while maintaining a high satisfaction of the patient time window and synchronization constraints. Indeed, in the case of hard time window, the memetic algorithm maintains a low deviation ratio compared to the best-known results. It also provides 7 new best solutions for the benchmark including 2 optimal solutions. In the case of soft time window, the memetic algorithm provides efficient solutions compared to the commercial solver Gurobi. Overall, it performs much better in the case of large instances.

In general, the memetic algorithm shows very good performance on the test instances, whether the home health care problem is considered with hard or soft time window and synchronization constraints, several home health care offices or various caregiver qualifications.

In future works, we plan to focus first on the workload balance between the different staff members in order to obtain a fair planning. In a later time, we project to develop an efficient exact resolution method on the same problem, beginning with providing a lower bound using Lagrangian relaxation method.

Finally, other real-life characteristics could be addressed in the model such as patient preferences for a given caregiver or blood samples delivery to a medical laboratory at the end of the route.

Acknowledgment

This work is supported by the Franche-Comté region in France (research project OptimHaD) under grant 2015C-4949.

References

[1] O. Bräysy, M. Gendreau, Vehicle routing problem with time windows, Part II: Metaheuristics, Transp. Sci. 39 (1) (2005) 119–139.
[2] G. Laporte, The vehicle routing problem: An overview of exact and approximate algorithms, European J. Oper. Res. 59 (3) (1992) 345–358.
[3] D. Bertrand, Les services de soins infirmiers à domicile en 2008, Etudes Et rèsultats 739 (2010). http://drees.social-sante.gouv.fr/IMG/pdf/er739.pdf.
[4] C. Fikar, P. Hirsch, Home health care routing and scheduling: A review, Comput. Oper. Res. 77 (2017) 86–95. http://www.sciencedirect.com/science/article/pii/S0305054816301848.
[5] S. Afifi, D.-C. Dang, A. Moukrim, Heuristic solutions for the vehicle routing problem with time windows and synchronized visits, Optim. Lett. 10 (3) (2016) 511–525.
[6] H. Allaoua, S. Borne, L. Létocart, R.W. Calvo, A matheuristic approach for solving a home health care problem, Electron. Notes Discrete Math. 41 (2013) 471–478.
[7] S. Bertels, T. Fahle, A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem, Comput. Oper. Res. 33 (10) (2006) 2866–2890. Part Special Issue: Constraint Programming.
[8] K. Braekers, R.F. Hartl, S.N. Parragh, F. Tricoire, A bi-objective home care scheduling problem: Analyzing the trade-off between costs and client inconvenience, European J. Oper. Res. 248 (2) (2016) 428–443.
[9] D. Bredström, M. Rönnqvist, Combined vehicle routing and scheduling with temporal precedence and synchronization constraints, Eur. J. Oper. Res. 191 (1) (2008) 19–31.
[10] E. Cheng, J.L. Rich, A home health care routing and scheduling problem, 1998. http://Citeseerx.Ist.Psu.Edu/Viewdoc/Summary.
[11] P. Eveborn, P. Flisberg, M. Rönnqvist, Laps carean operational system for staff planning of home care, European J. Oper. Res. 171 (3) (2006) 962–976.
[12] G. Hiermann, M. Prandtstetter, A. Rendl, J. Puchinger, G.R. Raidl, Metaheuristics for solving a multimodal home-healthcare scheduling problem, CEJOR Cent. Eur. J. Oper. Res. 23 (1) (2015) 89–113.
[13] B. Issaoui, I. Zidi, E. Marcon, K. Ghedira, New multi-objective approach for the home care service problem based on scheduling algorithms and variable neighborhood descent, Electron. Notes Discrete Math. 47 (2015) 181–188.
[14] Y. Kergosien, C. Lenté, J.-C. Billaut, Home health care problem: An extended multiple traveling salesman problem, in: 4th Multidisciplinary International Conference on Scheduling: Theory and Applications, MISTA'09, Dublin (Irlande), 2009, pp. 10–12..
[15] E. Lanzarone, A. Matta, Robust nurse-to-patient assignment in home care services to minimize overtimes under continuity of care, Oper. Res. Health Care 3 (2) (2014) 48–58. Special Issue of the 2012 conference of the {EURO} working group Operational Research Applied To Health Services (ORAHS). http://www.sciencedirect.com/science/article/pii/S2211692314000046.
[16] R. Liu, B. Yuan, Z. Jiang, Mathematical model and exact algorithm for the home care worker scheduling and routing problem with lunch break requirements, Int. J. Prod. Res. 55 (2) (2017) 558–575.
[17] D.S. Mankowska, F. Meisel, C. Bierwirth, The home health care routing and scheduling problem with interdependent services, Health Care Manage. Sci. 17 (1) (2014) 15–30.
[18] M.S. Rasmussen, T. Justesen, A. Dohn, J. Larsen, The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies, European J. Oper. Res. 219 (3) (2012) 598–610.
[19] Y. Shi, T. Boudouh, O. Grunder, A hybrid genetic algorithm for a home health care routing problem with time window and fuzzy demand, Expert Syst. Appl. 72 (2017) 160–176.
[20] A. Trautsamwieser, P. Hirsch, Optimization of daily scheduling for home health care services, J. Appl. Oper. Res. 3 (3) (2011) 124–136.
[21] J. Wirnitzer, I. Heckmann, A. Meyer, S. Nickel, Patient-based nurse rostering in home care, Oper. Res. Health Care 8 (2016) 91–102. http://dx.doi.org/10.1016/j.orhc.2015.08.005. http://www.sciencedirect.com/science/article/pii/S221169231420049X.
[22] S. Yalçındağ, A. Matta, E. Şahin, J.G. Shanthikumar, The patient assignment problem in home health care: using a data-driven method to estimate the travel times of care givers, Flexible Serv. Manuf. J. 28 (1–2) (2016) 304–335.
[23] M. Drexl, Synchronization in vehicle routinga survey of VRPs with multiple synchronization constraints, Transp. Sci. 46 (3) (2012) 297–316.
[24] N. Triki, T. Garaix, X. Xie, A two-phase approach for periodic home health care planning, in: 2014 IEEE International Conference on Automation Science and Engineering, CASE, 2014, pp. 518–523. http://dx.doi.org/10.1109/CoASE.2014.6899375.
[25] P. Moscato, On evolution, search, optimization, genetic algorithms and martial arts - Towards memetic algorithms, 1989.
[26] E. Alba, Parallel Metaheuristics: A New Class of Algorithms, John Wiley & Sons, 2005.
[27] J.-Y. Potvin, J.-M. Rousseau, A parallel route building algorithm for the vehicle routing and scheduling problem with time windows, European J. Oper. Res. 66 (3) (1993) 331–340.
[28] J.-Y. Potvin, T. Kervahut, B.-L. Garcia, J.-M. Rousseau, The vehicle routing problem with time windows part I: tabu search, INFORMS J. Comput. 8 (2) (1996) 158–164.
[29] I. Gurobi Optimization, Gurobi optimizer reference manual, 2016. http://www.gurobi.com.
[30] F. Gayraud, Problème de Tournées de Véhicules Avec Contraintes de Synchronisation Dans le Cadre de Structures de Maintien à Domicile. Formalisation Mathématique et Hybridation à Base de Métaheuristique, Université Blaise Pascal-Clermont-Ferrand II, 2015.