A heuristic approach to the task planning problem in a home care business

I Méndez-Fernández, S Lorenzo-Freire, I García-Jurado, J Costa, L Carpente

Abstract

In this paper, we study a task scheduling problem in a home care business. The company has a set of supervisors in charge of scheduling the caregivers' weekly plans. This can be a time-consuming task due to the large number of services they work with, as well as the need to consider user preferences, services required time windows and travel times between users' homes. Apart from that, it is also important to have a continuity of care, i.e., that users generally prefer not to have their caregiver changed.

This problem involves both route planning and employee task planning, which are usually very challenging. We first propose to model it using integer linear programming methodology. Since the real instances that the company needs to solve are very large, we design a heuristic algorithm, based on the simulated annealing philosophy, that allows the company to obtain the caregivers' weekly schedules. Lastly, we check the algorithm's good performance, by comparing the solutions it proposes with those provided by the integer linear programming methodology, in small size problems, and we present a case study to confirm that the algorithm correctly solves real-like instances.

Keywords: home care, scheduling, simulated annealing, integer programming, operations research, operations management

1. Introduction

Home care services aim to help elderly, sick or dependent people in maintaining or improving their life quality, without having to leave their homes. People often feel more comfortable and are more satisfied if they can continue living in their houses, instead of moving into a nursing home or specialised centre. To this end, home care businesses need to employ nurses, or caregivers, who are in charge of visiting clients' homes when necessary. During these visits, the nurses must handle the medical and/or domestic tasks required by the clients (such as cooking meals, administering medication, housekeeping chores, attending medical appointments, bathing, hair washing, etc). These appointments are generally set by the clients, so the schedules of the nurses must be adapted to the time restrictions imposed by them.

In this paper, we study the problem that arises in Mayores S.L., a company that provides home care services in the city of A Coruña and in the neighbouring municipalities. For this purpose, the company has a staff of caregivers who are in charge of assisting the clients in their homes. Each one of them has a schedule that specifies the period of time in which the visits to each client must occur. The company pays

its caregivers according to their working day, which begins at the moment the first planned service starts and ends when they complete the last one. Moreover, the longest rest of each day will not be paid to the caregiver, as long it reaches a certain duration.

Mayores wishes to have an expert decision-making system to obtain an automatic schedule for the caregivers weekly plan (specifying the services to be performed, the routes to be followed and the schedules to be upheld), in such a way that all the needs of the clients are covered. This automatic planning tool should take into account users' preferences and help reducing the total time worked, as much as possible. In addition, according to the requirements of the company, the tool should not excessively modify the previous caregivers' schedules.

This work is addressed within the so-called Home Health Care Scheduling Problems (a review is available in Fikar and Hirsch [1]), which are a combination of vehicle routing and scheduling problems (a thorough review of this combination of problems can be found in Paraskevopoulos et al. [2]). Although, as far as we know, our problem has not been previously studied, there are some home health care scheduling problems in the literature that share common points with it.

One of the most important characteristics of the problem presented by Mayores is the continuity of care, which means that it is desirable not to change the nurse (or nurses) who attends a patient. Some works in the literature study problems that present this characteristic. For example, in Bachouch et al. [3], the home care problem is solved in an optimal way that minimizes the total distances travelled by the nurses, and the schedules are such that they abide by patient and nurse constraints, in particular the fact that during the week the patients must be attended by the same person. This work differs from our problem because it considers that each user can only require one service per day and that the supervisors are in charge of deciding in which days the users' services will be conducted. Cappanera and Scutellà [4] study the case where the clients can only be attended by a maximum number of nurses while also considering that the nurses' maximum workload can be exceeded, the fact that the skills are not hierarchical, and that there may be time windows associated with the users. This problem is not like the one of Mayores, since in Cappanera and Scutellà [4] the breaks are not relevant at all. Moreover, they consider that the visits are not assigned to one specific day (instead, there are a number of services that must be performed during the week). Another home care problem with continuity of care is studied in Carello and Lanzarone [5]. In this case, the users, depending on their needs, may require hard continuity care, partial continuity care or not require continuity of care. They propose a robust assignment model, considering that the amount of work required by users is uncertain, in such a way that the schedules obtained minimize the costs due to reassignments of patients and those produced by nurses overtime. This problem is quite different from the one we are studying, since it involves uncertainty and does not work with the services' time windows.

The breaks play a very important role in our problem, since when calculating the daily cost of each caregiver the company will discount the largest break over two hours, if any. A similar case is considered

in Rest and Hirsch [6], which considers that nurses can work in two shifts, morning and afternoon, given that they have a break of a certain duration in their workday. But this problem also considers the fact that nurses make use of public transportation to travel from one home to another, and therefore, it addresses time dependencies and several means of transport and does not consider the continuity of care. Thus, this problem is different from the one presented by Mayores.

The users' satisfaction is of the utmost importance for Mayores, so it is advisable to consider their opinion when appointing the caregivers that will attend them. This characteristic is also considered in the literature. Duque et al. [7] present a home care problem in which two objectives are optimized, the service level (which comprises satisfying users' and nurses' preferences) and the total distance travelled. That work presents aspects that are not considered in our problem, such as the fact that the number of nurses assigned to a user varies according to the amount of services she[1] requires. In Braekers et al. [8], the objective is to minimize two functions, the total costs (produced by travel time and nurses' overtime) and the penalties (for assigning nurses to patients who have other preferences and for performing services outside their preferred starting times). It differs from the Mayores' problem because it takes into account the preferred service starting times declared by the patients. Another home care problem is studied in Eveborn et al. [9], where a set partitioning model is considered. They consider an optimization problem with a weighted objective function. This objective function includes several elements, such as the travel time, the scheduled hours, the preferences, some penalties, etc. This work is not identical to the one proposed by Mayores since in this case there are services that must be conducted by several nurses. In Misir et al. [10], a home care scheduling problem is studied where the solutions must minimize travel and idle times, while considering the fact that users may have preferences, that nurses can start their working days in different locations and that nurses could use different means of transport to travel between clients.

Several methods have been implemented in the literature to solve home care problems. Duque et al. [7] propose a two-stage solution strategy: first, the assignment of patients to nurses is obtained (optimizing the service level), and after that, the order in which nurses should visit them is established (to reduce the travelled distances). The problem presented in Braekers et al. [8] is solved in two phases: in the first one, they obtain the routes that the nurses will follow, and in the second one, they calculate the routes' schedule using a metaheuristic approach (multi-directional local search combined with large neighbourhood search). A two-stage solution approach for a home care problem is proposed in Nickel et al. [11]; first, they use a heuristic algorithm to obtain a feasible solution, and after that, they improve this solution by applying an adaptive large neighbourhood search procedure. They also consider the case in which one wants to work with a basic plan (the master schedule problem) that is then modified to solve different short-term changes that may occur (the operational planning problem). A hyper-heuristic method, that is, performing

[1] In this paper we use the female gender in a generic way. In any case, most of the caregivers in Mayores are women.

a search over a set of heuristics, is used in Misir et al. [10]. In Eveborn et al. [9], the home care problem is solved by implementing a repeated matching algorithm, and in Rasmussen et al. [12], the problem is solved using dynamic column generation in a branch-and-price framework. Finally, Rest and Hirsch [6] solve the problem using the tabu search method, obtaining schedules that minimize shift durations, overtime, number of afternoon shifts and overqualification of nurses.

Although there is a wide literature on planning in home care problems, the specific properties of the scheduling issues addressed by Mayores made them an unexplored research problem that will be treated in this paper. These specific properties are: the largest gap in the caregivers' working hours is unpaid, the services have time windows, and the different satisfaction levels between caregivers and users should be considered. We will discuss them all further in the next section. To solve the scheduling problem under study, we have modelled it as an integer linear programming problem. However, due to the fact that the real instances presented by the company result in a large number of variables and constraints, it is not possible to obtain their optimal solutions. Therefore, we have designed a heuristic algorithm, based on the simulated annealing method, introduced in Kirkpatrick et al. [13], that allows us to obtain good solutions in a reasonable computational time. Our algorithm, following the guidelines given by Mayores, provides plans that address the contemplated incidents and schedules the caregivers weekly plans, starting from an initial solution. The plans are obtained in such a way that they maximize the satisfaction levels between caregivers and customers, minimize the time lost between visits, do not modify in excess the previous schedules and ensure that visits are conducted within the clients' required time windows.

This paper is organized as follows. We introduce the concepts necessary to correctly describe the problem under study in Section 2. In Section 3, we present the integer linear programming model. The heuristic algorithm we designed is described in Section 4. We discuss the computational results obtained, to establish the algorithm's good performance, in Section 5. Finally, we present the conclusions in Section 6.

2. The problem

The aim of this work is to provide Mayores with a decision support system that allows it to obtain schedules in an automatic way. In this section, we introduce the scheduling problem to be solved.

Mayores attends a set of *users*, people who need professional help to conduct several tasks (either domestic chores or health care duties) and ask the company for help to improve their quality of life. These users require the realization of a number of *services*, the occasions during which the company's employees must visit the users' homes. For each one of these services its *duration* is known[2], and the user must specify the *day* of the week during which it must be completed, its *time frame* (morning, noon, afternoon or evening) and the *required time window* (within which the service must be completed).

[2]Sometimes it is specified by the user, sometimes it is estimated by the company based on its experience in similar services.

To provide its services, Mayores has a set of *caregivers* who are responsible for visiting the users' homes to perform several tasks (cleaning chores, cooking meals, helping maintain personal hygiene, etc.). Every caregiver has a *contract* that specifies the maximum number of hours she can work during the week. These contracts generate a wide variety of maximum permissible working hours, which can be extended or reduced, according to the needs of the company and its users, and in agreement with the caregivers.

Thus, each caregiver has a weekly *work plan* in which the following are specified: (i) the services she must perform, (ii) the hours during which the services must be conducted (always upholding the required time windows established by the users), (iii) the travel times between consecutive users and (iv) the breaks she has during her workday. The weekly work plan is updated every week. Every caregiver gets paid according to her work plan, considering that the greatest break she has in her workday (if its duration is equal to or greater than two hours) will not be remunerated.

When assigning caregivers to users, it is important to consider the users' opinions. This is due to the fact that if they are pleased with the caregiver who is attending them, they will usually prefer to continue with her. However, in case of conflicts, it is advisable to change the caregiver appointed to the user. Therefore, we work with a list of *affinity levels* that specify how suitable it is to assign a caregiver to each user. The affinity levels considered are as follows:

Level 5 This is the maximum level of affinity considered; it is reached when the caregiver is assisting the user on a continuous basis and in a satisfactory way.

Level 4 This level of affinity occurs when the caregiver assisted (sporadically or continuously) the user in the past but ceased to do so, not because of dissatisfaction but because of other organizational reasons.

Level 3 This level of affinity is reached when the caregiver has some type of essential attribute needed to serve the user, such as the ability to administer medication or knowledge to use a certain type of instrument, but has not ever assisted the user.

Level 2 This level is the average affinity. It is reached when the caregiver and the user have not interacted in the past, and the user does not require any special characteristic from the caregiver.

Level 1 This level of affinity is reached when the caregiver who assists (or was assisting) a user receives some type of mild complaint from the user. It is therefore advisable to replace the caregiver that attends the user, although she could assist the user as a last resort.

Level 0 This level of affinity arises on occasions when a user has banned a caregiver because there have been serious complaints or confrontations, and consequently the user does not want that caregiver to attend her in the future. It also arises when the caregiver does not have the essential expertise needed to serve the user.

To conduct the task of planning the weekly days of the caregivers, the company employs a number of *supervisors*, each of whom has been assigned a set of users and caregivers. These supervisors are in charge of organizing the work plans of the caregivers in such a way that all the needs of the users assigned to them are met. In other words, they plan the services required by the users, assigning them to the best caregiver available and specifying the schedule during which the services will be conducted.

The company's objective is to obtain the schedule of the caregivers automatically through the use of a computer application. It should be noted that the company does not intend to eliminate the supervisors (because they have many other administrative tasks), but its goal is to facilitate their work by providing them with an automatic scheduling tool.

The schedules must satisfy the constraints of both users and caregivers, while optimizing the following objectives to minimize the schedules' costs:

a. The work plan should maximize the affinity between the users and the caregivers attending them.
b. The work plan should minimize the time wasted between consecutive services (including travelling time).

Mayores continuously operates with a weekly plan for its caregivers. However, the plan is updated before each week starts if there are incidents reported for that week. The possible incidents considered are the following:

Registration of a new user When a new user is added to the system, it is necessary to schedule the services she requires.

Discharge of a user When a user is discharged from the system, it is necessary to remove her services from the schedules of the caregivers who attended her.

Increasing services When a user already in the system requires more services, these new services should be scheduled.

Decreasing services When a user who is already in the system requires the reduction of the services to be performed, these surplus services must be eliminated from the schedules of the caregivers who provide them.

Alteration of parameters When a user modifies any of her services' parameters (duration, time windows, day of the week, etc.), it is necessary to re-schedule those services to adapt them to their new characteristics.

3. The integer programming model

According to the previous section, it is clear that we are facing a multi-objective scheduling problem. Next, we model it as an integer programming problem. First, we define the parameters[3] and variables that are involved in the problem, and after that, we present the model that was designed for its resolution.

The *parameters* involved in formulating the model are as follows.

- $Q = \{1, ..., 7\}$ is the set of days.
- $N = \{1, ..., n\}$ is the set of caregivers.
- $M = \{1, ..., m\}$ is the set of time periods. Each time period lasts 5 minutes.
- $\overline{M} = \{m_0, m_1, ..., m_7\}$ is the set of time periods that mark the end of each day considered, where $m_1, ..., m_7 \in M$, $m_0 = 0$ is a dummy period and $m_7 = m$.
- $S = \{1, ..., s\}$ is the set of services. We also consider two dummy services, 0 the initial dummy service and $s+1$ the ending dummy service. Therefore, we also have the following sets of services: $S^0 = S \cup \{0\}$, $S^1 = S \cup \{s+1\}$ and $S^{01} = S \cup \{0\} \cup \{s+1\}$.

  Since each service should be carried out in a particular day, we also define the sets of services at day q by S_q. Similarly, $S_q^0 = S_q \cup \{0\}$, $S_q^1 = S_q \cup \{s+1\}$ and $S_q^{01} = S_q \cup \{0\} \cup \{s+1\}$.
- $a_{1k} \in \mathbb{N}$ is the duration of service $k \in S$.
- $a_{2k} \in M$ is the starting time of the required time window of service $k \in S$.
- $a_{3k} \in M$ is the ending time of the required time window of service $k \in S$.
- $f_{ik} \in \{0, 1, 2, 3, 4, 5\}$ $\forall i \in N$, $\forall k \in S$, is the affinity level[4] between the caregiver $i \in N$ and the service $k \in S$.
- $T = (t_{kl})_{s \times s}$ denotes the travel time between services, that is, t_{kl} is the travel time between service $k \in S$ and service $l \in S$.
- hs_i is the maximum time that caregiver $i \in N$ is allowed to work in a week, and hd_{iq} is the maximum time that caregiver $i \in N$ is allowed to work in day $q \in Q$.
- $E, C, D \in \mathbb{N}$ are fixed values, large enough to satisfy the constraints where they are used.

[3]Note that to model the problem under study, instead of working with the users, we work with the services they require (which can be more than one per day).

[4]To simplify the notation, we define the affinity levels between caregivers and services (instead of users). Note that the resulting problem is, in fact, more general than the one presented by Mayores. In practice, we associate to each caregiver and service the affinity level between the caregiver and the user requiring the service.

The *decision variables* considered are the following:

- For all $i \in N$, $k \in S^0$, $j \in [a_{2k}, a_{3k} - a_{1k}]$ and $l \in S^1$, such that there exists $q \in Q$ with $k, l \in S_q^{01}$,
$$x_{ijkl} = \begin{cases} 1, & \text{if caregiver } i \text{ starts service } k \text{ in time period } j \text{ and the next service} \\ & \text{is service } l; \\ 0, & \text{otherwise.} \end{cases}$$
- For all $i \in N$, $q \in Q$, $k, l \in S_q$ and $p \in S_q^1$, where k, l, and p are different services,
$$y_{iqklp} = \begin{cases} 1, & \text{if the largest rest of caregiver } i \text{ on day } q \text{ has a duration greater than 0 and} \\ & \text{is between services } k \text{ and } l, \text{ after she provides service } p; \\ 0, & \text{otherwise.} \end{cases}$$
- For all $i \in N$ and $q \in Q$,
$$\overline{y}_{iq} = \begin{cases} 1, & \text{if the largest rest of caregiver } i \text{ on day } q \text{ has a duration of 0;} \\ 0, & \text{otherwise.} \end{cases}$$
- For all $i \in N$ and $q \in Q$,
$$z_{iq} = \begin{cases} 1, & \text{if the largest rest for caregiver } i \text{ on day } q \text{ lasts 2 hours} \\ & \text{or more;} \\ 0, & \text{otherwise.} \end{cases}$$
- For all $i \in N$ and $q \in Q$, $\mu_{iq} \geq 0$ represents the largest rest for caregiver i in day q.
- For all $i \in N$ and $q \in Q$, $d_{iq} = \begin{cases} \mu_{iq}, & \text{if } \mu_{iq} \text{ has a duration equal to or greater than 2 hours;} \\ 0, & \text{otherwise.} \end{cases}$

The integer programming problem is

$$\max \sum_{i \in N} \sum_{q \in Q} \sum_{k,l \in S_q} \sum_{j \in [a_{2k}, a_{3k} - a_{1k}]} f_{ik} x_{ijkl} \tag{1}$$

$$\min \sum_{i \in N} \sum_{q \in Q} \left(\sum_{k \in S_q^0} \sum_{j \in [a_{2k}, a_{3k} - a_{1k}]} (j + a_{1k}) \times x_{ijks+1} - \sum_{l \in S_q^1} \sum_{j \in [m_{q-1}+1, m_q]} j \times x_{ij0l} - d_{iq} \right) \tag{2}$$

subject to

$$\sum_{i \in N} \sum_{l \in S_q^1 \setminus \{k\}} \sum_{j \in [a_{2k}, a_{3k} - a_{1k}]} x_{ijkl} = 1 \qquad \forall k \in S_q, \forall q \in Q \tag{3}$$

$$\sum_{i \in N} \sum_{k \in S_q^0 \setminus \{l\}} \sum_{j \in [a_{2k}, a_{3k} - a_{1k}]} x_{ijkl} = 1 \qquad \forall l \in S_q, \forall q \in Q \tag{4}$$

$$\sum_{l \in S_q^1} \sum_{j \in [m_{q-1}+1, m_q]} x_{ij0l} = 1 \qquad \forall i \in N,\ \forall q \in Q \qquad (5)$$

$$\sum_{k \in S_q^0} \sum_{j \in [a_{2k}, a_{3k} - a_{1k}]} x_{ijks+1} = 1 \qquad \forall i \in N,\ \forall q \in Q \qquad (6)$$

$$x_{ijkl} \leq \sum_{v=j+a_{1k}+t_{kl}}^{a_{3l}-a_{1l}} \sum_{\substack{r \in S_q^1: \\ r \neq k}} x_{ivlr} \qquad \forall i \in N,\ \forall j \in [a_{2l}, a_{3l} - a_{1l}], \forall k, l \in S_q,\ \forall q \in Q \quad (7)$$

$$x_{ijkl} = 0 \qquad \forall i \in N,\ \forall k, l \in S_q,\ \forall j \in [a_{2k}, a_{3k} - a_{1k}] \ : \ j + a_{1k} + t_{kl} > m_q, \forall q \in Q \qquad (8)$$

$$x_{ij0l} \leq \sum_{p \in S_q^1} x_{ijlp} \qquad \forall i \in N,\ \forall j \in [a_{2l}, a_{3l} - a_{1l}]\ \forall l \in S_q, \forall q \in Q \qquad (9)$$

$$x_{ijkl} \leq f_{ik} \qquad \forall j \in [a_{2k}, a_{3k} - a_{1k}],\ \forall l \in S_q^1,\ \forall i \in N,\ \forall k \in S_q, \forall q \in Q \qquad (10)$$

$$\sum_{j \in [a_{2k}, a_{3k} - a_{1k}]} \sum_{k \in S_q^0} (j + a_{1k}) \times x_{ijks+1} - \sum_{j \in [m_{q-1}+1, m_q]} \sum_{l \in S_q^1} j \times x_{ij0l} - d_{iq} \leq h d_{iq} \qquad \forall q \in Q,\ \forall i \in N \qquad (11)$$

$$\sum_{q \in Q} \left(\sum_{j \in [a_{2k}, a_{3k} - a_{1k}]} \sum_{k \in S_q^0} (j + a_{1k}) \times x_{ijks+1} - \sum_{j \in [m_{q-1}+1, m_q]} \sum_{l \in S_q^1} j \times x_{ij0l} - d_{iq} \right) \leq h s_i \quad \forall i \in N \qquad (12)$$

$$\mu_{iq} \leq C(1 - \overline{y}_{iq}) \qquad \forall i \in N,\ \forall q \in Q \qquad (13)$$

$$\mu_{iq} + D(1 - \sum_{j \in [a_{2k}, a_{3k} - a_{1k}]} x_{ijkl}) \geq \left(\sum_{j \in [a_{2l}, a_{3l} - a_{1l}]} j \times x_{ijlp} - \sum_{j \in [a_{2k}, a_{3k} - a_{1k}]} (j + t_{kl} + a_{1k}) \times x_{ijkl} \right)$$
$$\forall i \in N,\ \forall k, l \in S_q,\ \forall p \in S_q^1, \forall q \in Q \qquad (14)$$

$$\mu_{iq} + D(1 - \sum_{j \in [a_{2k}, a_{3k} - a_{1k}]} x_{ijkl}) \leq \left(\sum_{j \in [a_{2l}, a_{3l} - a_{1l}]} j \times x_{ijlp} - \sum_{j \in [a_{2k}, a_{3k} - a_{1k}]} (j + t_{kl} + a_{1k}) \times x_{ijkl} \right) + C(1 - y_{iqklp})$$
$$\forall i \in N,\ \forall k, l \in S_q,\ \forall p \in S_q^1, \forall q \in Q \qquad (15)$$

$$\sum_{k \in S_q} \sum_{l \in S_q} \sum_{p \in S_q^1} y_{iqklp} + \overline{y}_{iq} = 1 \qquad \forall i \in N,\ \forall q \in Q \qquad (16)$$

$$\frac{\frac{\mu_{iq}}{2 \cdot 12} - 1}{E} < z_{iq} \leq \frac{\mu_{iq}}{2 \cdot 12} \qquad \forall i \in N,\ \forall q \in Q \qquad (17)$$

$$d_{iq} \leq E z_{iq} \qquad \forall i \in N,\ \forall q \in Q \qquad (18)$$

$$d_{iq} \leq \mu_{iq} \qquad \forall i \in N,\ \forall q \in Q \qquad (19)$$

$$d_{iq} \geq \mu_{iq} - E(1 - z_{iq}) \qquad \forall i \in N,\ \forall q \in Q \qquad (20)$$

The objective function is a lexicographical one which, first (1) maximizes the affinity levels when assigning caregivers to services and second (2) minimizes the total time worked (except for the largest break of each caregiver's workday in case its duration is greater than, or equal to, two hours).

Constraint (3) guarantees that every service is performed within its required time window. Constraint (4) requires that each service can only have one previous service. Constraints (5) and (6) assure that each caregiver starts her working day with the initial dummy service and finishes it with the ending dummy service. Constraints (7) to (9) guarantee that the starting times of two consecutive services are feasible, that is, the time between them is sufficient to cover the time devoted to provide the first service and the time spent in travelling to the second one. The next constraint (10) assures that no service will be assigned to caregivers when their affinity level is 0. Constraints (11) and (12) ensure that the scheduled working hours for each caregiver do not surpass their daily and weekly maximum number of working hours allowed. Constraint (13) guarantees that the largest break in the working day of each caregiver will have a duration of, at least, 0. Constraints (14) and (15) assure that, to obtain the largest break, all the possible breaks in the working day of each caregiver are considered. Constraint (16) ensures that for every caregiver, on each working day, only a single breaks will be considered as the largest one. Finally, constraints (17) to (20) are necessary to determine whether the largest breaks of every caregiver, on each working day, have a duration equal to or greater than two hours.

4. The heuristic approach

The problem presented by Mayores has a very large computational complexity, as shown in the previous section. Therefore, it will be necessary to address it using heuristic techniques, to obtain acceptable solutions in short computational times. In this section, we describe the algorithm that we designed to solve this problem. It is based on the simulated annealing method and consists of scheduling the considered services while optimizing the working days of the caregivers (so that the plans are feasible, the caregivers' breaks are minimized, and all the services are correctly scheduled).

Figure 1 shows the process we follow to generate the schedules. The algorithm can be divided into two stages: an initial one in which the main information is extracted, and a second one in which the schedule is optimized.

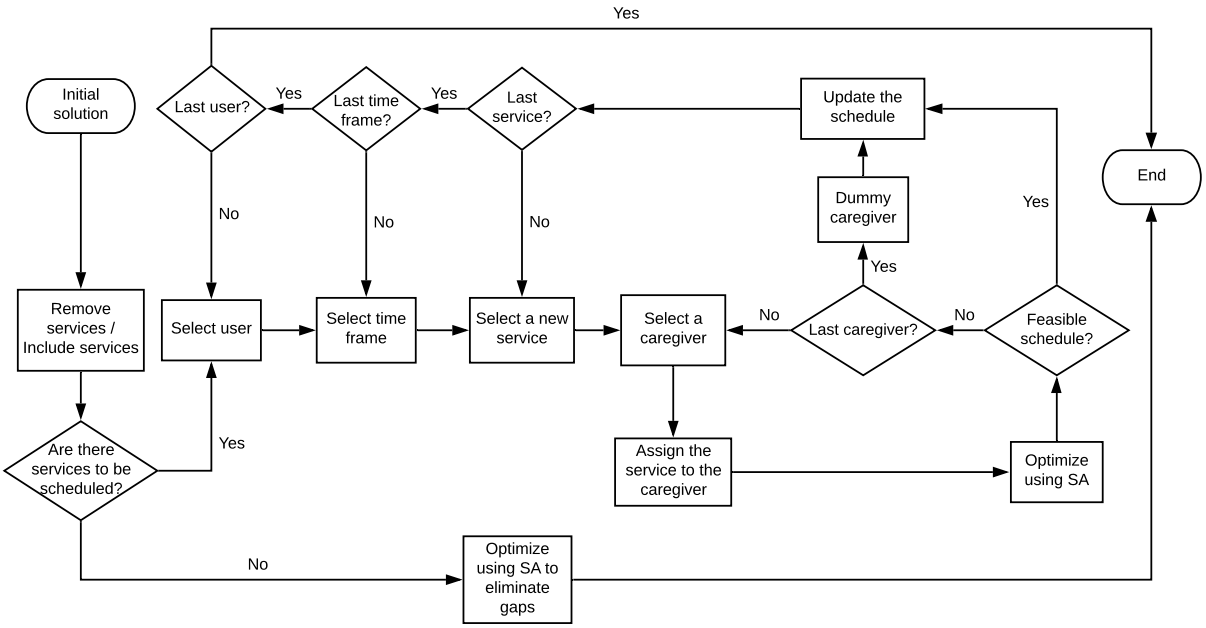

Figure 1: Diagram of the algorithm.

4.1. Initial phase

The algorithm begins with an initial solution: the previous schedules of the considered caregivers. From this schedule, we must exclude the services that have to be deleted (departures or reduction of services) and also select all those that need to be scheduled (registration, increase of services and alterations).

If no services must be scheduled, we optimize the solution applying the simulated annealing method, to remove any break that may exist, after which the algorithm ends.

4.2. Service scheduling

This second phase of the algorithm is performed when there are services to be scheduled. Here we establish the caregiver who must perform each of these services and the schedule during which they must be conducted to minimize costs and obtain feasible schedules.

The first step in this phase is to select the service to be scheduled and to obtain a list of the available caregivers[5], sorted from best to worst, considering the following elements:

[5]By available caregivers we mean that they do not surpass their daily maximum number of hours allowed.

a. The affinity level between the caregiver and the considered user.

b. The difference between the time worked by the caregiver and her maximum time allowed.

c. The average travel time between the considered user and all the users served by the caregiver.

Then, we assign the selected service to the best caregiver available, scheduling it so that the least overlap with other services of this caregiver is produced and, then, so that the minimal break with other services of this caregiver is produced. After that, as shown in Figure 1, we try to optimize the schedules applying the simulated annealing method. If we find an optimum schedule of the service, we go to the next one; if we conclude that there is not a feasible schedule of the service with the current caregiver, we go to the next available caregiver and repeat the process. If we have already tested all of the available caregivers, we assign the service to a dummy caregiver (to allow the supervisors to easily know which services could not be properly scheduled). When the algorithm ends with a solution that includes n dummy caregivers it is indicating to the company that it cannot find a feasible solution unless they hire n new caregivers.

4.3. The optimization

As previously explained, it is sometimes necessary to perform an optimization process to eliminate breaks and/or overlaps from the caregivers' schedules. To do this, we use a modified version of the simulated annealing method to fully adapt it to the problem under study. The idea of the simulated annealing method comes from the analogy between physical annealing of solids and combinatorial optimization problems. To implement this method, it is necessary to define the problem in terms of a solution space with a neighbourhood and a cost function. The algorithm starts with an initial solution and then moves randomly through the solution space. To do this we select a number l of movements to perform; they are selected at random according to a probability distribution over all the possible ones. Each of the l movements is used to produce a neighbour and, then, between all of them we pick the one with the minimum cost: which is the new solution obtained in this iteration. If the new solution provides a better result, in terms of the cost function, than the current solution at hand (after the past iterations) we choose the new solution. Otherwise, even if it provides a worse result, there is still a chance that the new solution will be accepted (depending on the temperature and the increment of the objective function; see [13] for more details on the simulated annealing algorithm), but in this case we will most probably reject the new solution and keep the current one.

The probability distribution used to select the movements is updated after each iteration. At the beginning it is the uniform discrete distribution and, in every iteration, the type of movement leading to the best neighbours is given a heavier weight in the distribution.

In our approach, the cost function used during the simulated annealing optimization considers the lexicographical order applied on these criteria:

a. The total time of overlap in the schedule.

b. The time lost between services and the time spent travelling between services.

4.3.1. The movements

At each simulated annealing iteration, we implement movements to retrieve the neighbours of the solution under consideration. These movements can be divided into two classes: the basic movements, which are the simplest ones and tend to generate the best solutions, and the elaborate ones, which arise from merging and/or modifying basic movements, and are not usually used as much as the others but, sometimes, are essential to obtain better solutions.

The basic movements we designed are the following:

First movement. This movement consists in modifying the schedule during which a service is performed by moving it forward or backward.

Step 1. We select, at random, one service and the number of minutes that it will be delayed or advanced.

Step 2. We modify the schedule in which the selected service is delayed if the number of minutes selected is positive or forwarded if a negative number is selected.

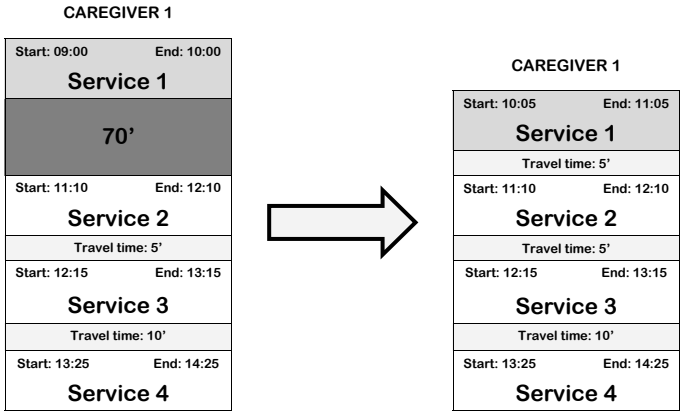

Figure 2: Example of the first movement.

Example 4.1 Suppose that a caregiver conducts four services, as shown in Figure 2, and has a 70-minute break between service 1 and service 2. Using movement 1, we can remove this break, since delaying service 1 as much as possible (always in accordance with the travelling time to service 2) results in the elimination of the break from the caregiver's schedule.

Second movement. This movement consists of advancing or delaying a set of consecutive services.

Step 1. We select at random one service, the number of minutes that the services' schedules will be delayed or advanced.

Step 2. If the direction is next, we select the set of services following the given service, and if the direction is previous, we select the set of services preceding the given service. Note that, in this step, we

select only a set of consecutive services between which there is no break with a duration equal to or greater than two hours.

Step 3. We modify the schedule in which the services of the selected set are performed, delaying them the given minutes (if the quantity is positive) or advancing them the given minutes (if the quantity is negative).

Example 4.2 Suppose that a caregiver has four services with a 65-minute break (Figure 3). If we implement the second movement, delaying services 1 and 2 the same number of minutes (always considering the travel time between service 2 and service 3), the break between services 2 and 3 is eliminated.

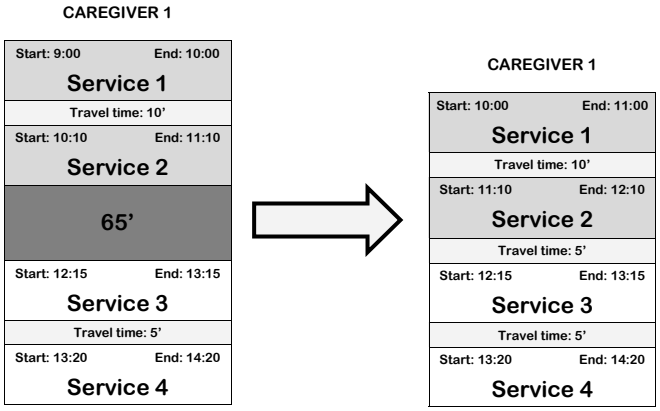

Figure 3: Example of the second movement.

Third movement. This movement comprises exchanging the starting times for two given services assigned to the same caregiver on a given day. In this movement, there is also the option of applying the first movement after having conducted the exchange.

Step 1. Two different services are chosen, such that they are both performed by a single caregiver during the same day.

Step 2. The starting times of the selected services are exchanged, and their ending times are adapted to their new starting times.

Step 3. We randomly decide if the first movement will be applied to the selected services, and if it is the case, the schedules of both services are altered by performing this movement.

Example 4.3 Let us suppose a caregiver performs four services (Figure 4) with a 35-minutes break between service 2 and service 3. If we use movement 3, exchanging the starting times between services 2 and 4, we manage to remove the break. The reason for this is that service 4 has a duration of 90 minutes, while service 2 lasts only 60 minutes; therefore, this extra duration of service 4 fills the break that the caregiver had in her schedule.

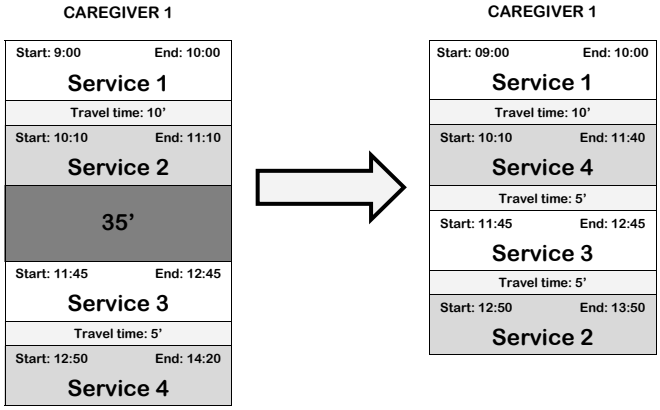

Figure 4: Example of the third movement.

Fourth movement. This movement consists of replacing the caregiver who conducts a given service.

Step 1. We select one service at random and one caregiver (other than the one who performs this service).

Step 2. The selected service is assigned to the new caregiver, maintaining the hours in which it must be conducted.

Example 4.4 In this example, we consider two caregivers (Figure 5). In the initial schedule of caregiver 1, we see that service 3 is isolated, because there are breaks before and after it. If we use movement 4 such that service 3 will now be conducted by caregiver 2, we observe that the break of caregiver 1 now has a duration of 135 minutes (therefore, the company is no longer required to pay her for it) and that assigning service 3 to caregiver 2 generates no breaks in her schedule.

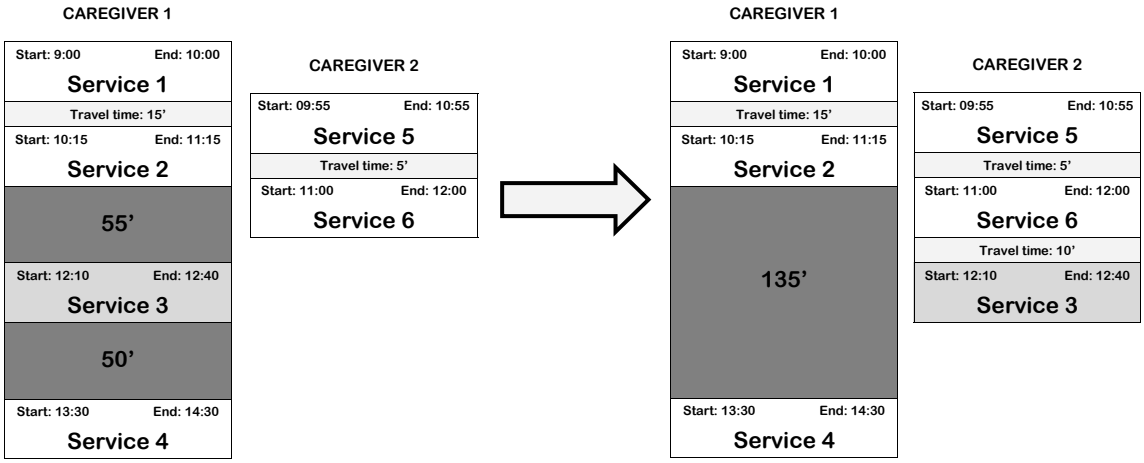

Figure 5: Example of the fourth movement.

The elaborated movements we considered in the simulated annealing method are:

Fifth movement. This movement comprises exchanging the caregivers of two services that are performed by different caregivers on the same day.

Sixth movement. This movement consists of applying twice the second movement for a service: for the service and the ones before it, and then for the services after it.

Seventh movement. Given two sets of services, this movement exchanges the caregivers who conduct them. To correctly apply this movement, all the services in each set must be performed by the same caregiver.

5. Computational results

In this section, we present the computational results that allow us to evaluate the performance of the heuristic algorithm. To do this, first, we use the integer programming model and the heuristic algorithm to solve a battery of different sized instances, and then we show a real-life example for which we use the heuristic algorithm to solve some of the incidents the company addresses.

5.1. Comparison between the exact method and the heuristic algorithm

We now study the behaviour of the integer programming model and the heuristic algorithm by using them to solve a battery of instances, taking into account that there is no initial solution, which means that the two methods should provide a schedule from scratch. To have instances that correctly represent the reality of Mayores, our study is based on the company's available information. According to it, we established that the travel distances between users are 5, 10 or 15 minutes; the affinity levels between caregivers and users are 2, 3, 4 or 5; the services' durations are 30, 45, 50, 60, 70, 80, 90, 120 or 165 minutes; and the required time windows are such that the slack, in relation to the services' durations, can take values between 0 and 420 minutes.

We generated two types of instances: small ones (instances 1 to 30) and big ones (instances 31 to 45). On the one hand, for the small instances we begin with some basic ones that have only one caregiver and one day, which are Travelling Salesman Problems with our problem's particular characteristics. Then, to check that both the integer approach and the algorithm correctly solve the problem under study, we incremented the number of routes by varying the number of days (up to 2) and increasing the number of caregivers (up to 3) and services (up to 35) considered. On the other hand, for the big instances we increased the number of caregivers to 10 and the number of days to 7, which is more similar to the number of caregivers and days involved in real problems. In those instances we also incremented the number of services considered in each one of them (up to 150) to test the performance of the methods.

5.1.1. Exact method

In Table 1, we present the results of solving the battery of instances using the Gurobi Optimizer on an Intel Core i7-8700K CPU at 9.70 GHz, with a time limit of 90 minutes. The attributes considered for

Id	N	Q	S	Var	Constr	Time	Gap	A	W	Id	N	Q	S	Var	Constr	Time	Gap	A	W
1	1	1	6	3023	2051	0.52	0%	24	59	24	3	1	15	67479	83172	4.93	0%	105	197
2	1	1	7	3779	2748	1.56	0%	28	68	25	3	1	21	52026	68115	1.33	0%	102	221
3	1	1	7	3891	2860	0.77	0%	28	88	26	3	2	14	24816	17923	90.00	5540%	70	220
4	1	1	8	4411	3467	3.70	0%	32	79	27	3	2	19	15575	12077	90.00	-	-	-
5	1	1	8	4480	3563	27.50	0%	32	67	28	3	2	23	22688	18522	90.00	-	-	-
6	1	2	12	5269	3129	2.84	0%	24	136	29	3	2	28	33910	28909	90.00	-	-	-
7	1	2	13	28716	20501	90.00	10046%	65	213	30	3	2	35	55363	49193	90.00	-	-	-
8	1	2	14	7697	5274	90.00	27%	24	167	31	10	7	10	16570	8030	38.10	0%	50	192
9	1	2	16	7940	5285	12.61	0%	32	178	32	10	7	20	67150	32210	90.00	2251%	100	312
10	1	2	20	12266	9261	4.62	0%	40	139	33	10	7	30	128530	62210	90.00	11561%	150	444
11	2	1	10	8646	7242	1.15	0%	43	108	34	10	7	40	247640	123980	90.00	-	-	-
12	2	1	10	8690	7682	0.05	0%	50	98	35	10	7	50	432140	219420	90.00	20999%	250	702
13	2	1	12	12394	12190	0.08	0%	60	118	36	10	7	60	569810	199610	90.00	-	-	-
14	2	1	14	16130	16374	0.08	0%	70	155	37	10	7	70	682230	241670	90.00	-	-	-
15	2	1	14	16214	17718	0.60	0%	66	141	38	10	7	80	1283830	454250	90.00	-	-	-
16	2	2	12	8060	4218	1.36	0%	52	213	39	10	7	90	1315270	471050	90.00	-	-	-
17	2	2	20	20308	14522	2.63	0%	92	199	40	10	7	100	2093930	715130	90.00	-	-	-
18	2	2	20	23446	17462	0.26	0%	97	211	41	10	7	110	2384590	826970	90.00	-	-	-
19	2	2	22	22020	15706	0.15	0%	100	211	42	10	7	120	3067050	1061090	90.00	-	-	-
20	2	2	22	25380	18346	0.63	0%	99	333	43	10	7	130	3603690	1229150	90.00	-	-	-
21	3	1	12	15567	15249	0.05	0%	60	130	44	10	7	140	4786250	1631630	90.00	-	-	-
22	3	1	13	19911	19988	0.37	0%	57	148	45	10	7	150	6975130	2310470	90.00	-	-	-
23	3	1	14	38103	46185	4.73	0%	84	201										

Table 1: Computational study of the LP problem.

the generated instances are: identification number (Id), number of caregivers (N), number of days (Q) and number of services (S). Apart from that, we present the details of the exact resolution method like the problem size (number of variables and constraints), the minutes needed to solve the instances (taking into account that we let it run for a maximum time of 90 minutes), the gap obtained when the execution of the method is finished, the total affinity between services and caregivers given by the solution (A) and the caregivers' working time according to the obtained schedule (W).

5.1.2. Heuristic algorithm

The heuristic algorithm is based on the simulated annealing method, as shown in Section 4. We set the following parameters for its implementation.

- The initial temperature is $T_0 = 100$.
- The cooling function for iteration k is $T_k = T_0\beta^k$, with $\beta = 0.8$.
- The acceptance probability is $P(acceptance) = exp(-(f(s') - f(s))/T_k)$, s' is the new solution, and s is the previous one.
- The maximum number of iterations allowed is $n = 100$, and the maximum number of iterations allowed without improving the considered solution is $m = 10$.

Id	N	Q	S	Time	A	W	Id	N	Q	S	Time	A	W
1	1	1	6	0.05	24	59	24	3	1	15	0.70	105	197
2	1	1	7	0.06	28	68	25	3	1	21	0.47	102	221
3	1	1	7	0.03	28	88	26	3	2	14	0.09	70	212
4	1	1	8	0.05	32	79	27	3	2	19	0.12	95	260
5	1	1	8	0.05	32	67	28	3	2	23	0.16	115	320
6	1	2	12	0.27	24	136	29	3	2	28	0.12	140	370
7	1	2	13	0.07	65	194	30	3	2	35	0.13	175	469
8	1	2	14	0.68	24	167	31	10	7	10	0.08	50	192
9	1	2	16	0.93	32	178	32	10	7	20	0.08	100	312
10	1	2	20	0.01	40	139	33	10	7	30	1.10	150	434
11	2	1	10	1.29	43	108	34	10	7	40	0.10	200	573
12	2	1	10	0.10	50	98	35	10	7	50	0.13	250	697
13	2	1	12	0.25	60	118	36	10	7	60	0.17	199	842
14	2	1	14	0.16	70	155	37	10	7	70	0.24	348	946
15	2	1	14	1.17	66	141	38	10	7	80	0.28	399	1053
16	2	2	12	1.07	52	213	39	10	7	90	0.19	450	1241
17	2	2	20	0.59	92	199	40	10	7	100	0.32	499	1418
18	2	2	20	0.93	97	211	41	10	7	110	0.14	550	1529
19	2	2	22	0.99	100	211	42	10	7	120	0.52	600	1695
20	2	2	22	0.37	99	333	43	10	7	130	0.83	650	1804
21	3	1	12	0.17	60	130	44	10	7	140	0.67	700	2048
22	3	1	13	1.50	57	148	45	10	7	150	1.46	750	2288
23	3	1	14	0.22	84	201							

Table 2: Computational study of the SA algorithm.

- The size of the neighbourhood considered at each iteration is $l = 100$.

In Table 2 we present the results obtained by solving the battery of instances using the SA heuristic, which was implemented in Java and run on an Intel Core i7-7500U CPU at 2.70 GHz. In this table we have the number of caregivers (N), days (Q) and services (S) involved in the schedule, as well as the time the heuristic algorithm needed to solve them, the total affinity of the obtained schedule (A) and the expected working time of the caregivers according to the solution (W).

5.1.3. Comparison between the methods

In Tables 1 and 2, for the smaller instances (1 to 30), we can see that both methods correctly solve most of them (that is, both methods provide us with the optimal solution) in short CPU times. There are only a few exceptions to this. First, although in instances 7 and 26 Gurobi is able to reach the optimal value of the affinity objective function, this is not the case with the caregivers' working time (which is worse than the one provided by the heuristic algorithm). Second, in instance 8 both methods provide solutions with the same objective values but Gurobi is not able to guarantee optimality. Finally, in instances 27 to 30 the exact method cannot find a feasible solution in the given time, whereas the heuristic algorithm provides reasonable solutions in a short time.

Moreover, Gurobi just solves 4 of the largest instances (instances from 31 to 45), guaranteeing optimality only for instance 31. In all four cases, Gurobi is not able to improve the solutions obtained with the algorithm.

Apart from that, the heuristic algorithm is the only one of both methods that finds a feasible solution for the rest of the instances, obtaining them in short CPU times.

Therefore, we can conclude that our heuristic algorithm is capable of finding in short time solutions of equal quality (and even better in some cases) than those provided by the integer programming problem.

5.2. Real-life instance

Due to the fact that the previous instances do not fully represent the problems that the company needs to solve, we now present an example of a real-life instance where we want to solve some of the incidents described in Section 2. In Mayores, a supervisor can work with approximately 15 to 40 caregivers and 40 to 200 users but, in her day-to-day work, the incidents she must solve do not usually affect all of them. Therefore, she usually has to work with the schedule of a subset of the caregivers and users. Because of this, the example shown below is a good representation of a supervisor real problem.

In this example, we want to register two new users, discharge one user, and change the duration of the service of a user. It is important to mention that, for simplicity, we are showing only the schedule of one day of the week (but the algorithm works with the whole week), and for this day, each user only requires the execution of a single service. In Table 3, we present the incidents' attributes, that is, the user requiring them, the incident types, the services durations and their required time windows.

User	Incident	Duration	Available Time Window
26	Add service	30	9:30 - 12:00
27	Add service	30	11:30 - 13:30
09	Remove service	–	–
10	Service new duration	45	9:00 - 12:00

Table 3: Incidents.

The heuristic algorithm has been implemented in a software tool that was developed for Mayores, to allow the supervisors to automatically schedule the working days of the caregivers. In Figure 6, we use the software tool to present the initial schedule of the example under study. In this figure, each column represents a single caregiver's schedule, and each of the services she must complete is identified by a box (marked in green colour if the service is correctly scheduled) inside which we can see the name of the user who requires it, its required time window, and the time at which the service is currently scheduled.

In this example, we have 4 caregivers (columns) and 25 users (with one service for each of them). On the one hand, we can see that User 09's service is marked in red, which means that the user has to be discharged, and that User 10's service is marked with a warning signal because its previously scheduled duration differs from its new duration. On the other hand, it can be seen that Caregiver 02 has a break in her schedule with a duration of 2 hours and 15 minutes (marked in red), between the services of User 15 and User 16; this break is not included in her working day, so there is no need to eliminate it. Finally, the services of User 26 and User 27 are not currently assigned to any caregiver.

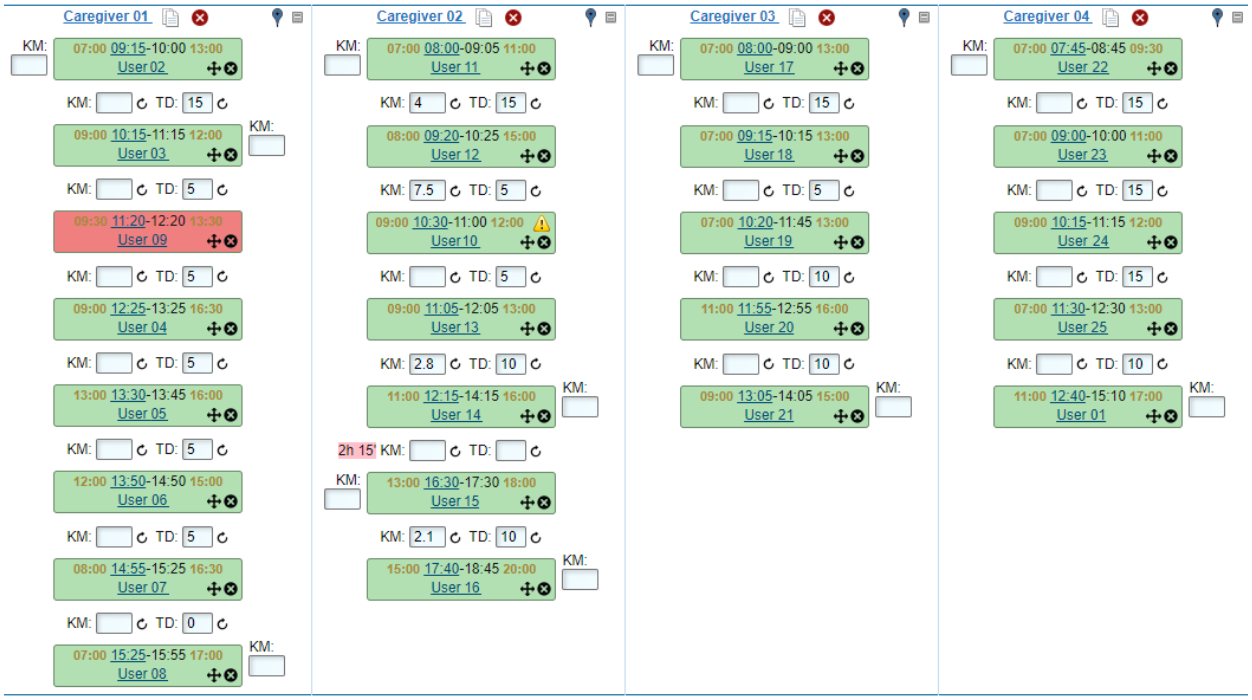

Figure 6: Initial schedule.

In Figure 7, we have the new schedule, obtained through the use of the heuristic algorithm in 1.63 minutes[6]. First, we can see that the services of User 26 and User 27 have been correctly scheduled: the one of User 26 was assigned to Caregiver 04, and the one of User 27 was assigned to Caregiver 03. Additionally, User 09's service was removed from the plan of Caregiver 01, and her other services' schedules were adapted to remove the break that was generated with the elimination of User 09's service. Lastly, User 10's service (originally assigned to Caregiver 03) was adapted to its new duration of 45 minutes, which leads to the delay of User 13's and User 14's services. In addition, we can state that the algorithm is very conservative, since it does not modify much the initial schedules.

Therefore, we have illustrated that our algorithm can be used to solve real-life problems and that the company can use it to schedule their caregivers' weekly work plans.

6. Conclusion

In this work, we have described a task planning problem proposed by a home care business that presents a collection of characteristics that had not been previously studied in the literature. To solve it, we have modelled it as an integer programming problem. However, the model allowed us to solve the problem

[6]Note that we executed the algorithm to solve the incidents for the whole week, not just for one day.

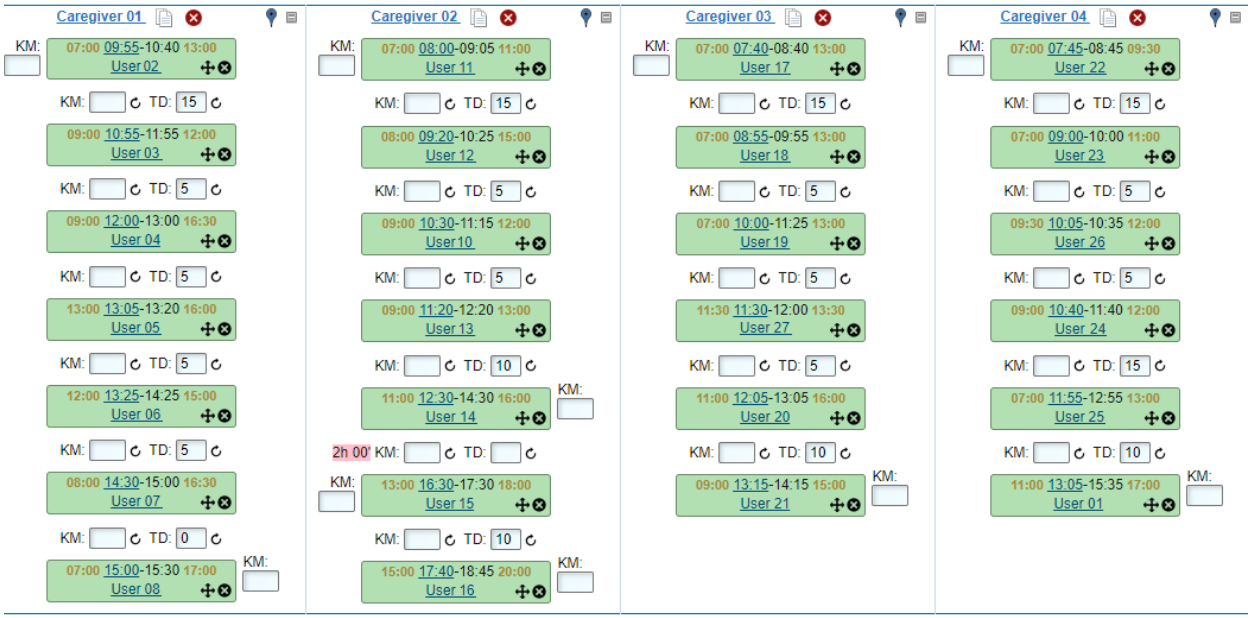

Figure 7: Final schedule.

optimally only for small-size instances. Since the real examples that the business needs to solve on a day-to-day basis are considerably large, we have also designed a heuristic algorithm, based on the simulated annealing method, to solve real problems in short CPU times. This algorithm was implemented in a software tool, to provide Mayores with a decision support system that will help them to obtain the caregivers' weekly schedules and to update them whenever necessary due to the occurrence of incidents. We checked the good performance of the heuristic algorithm by solving a set of small-sized instances and comparing the solutions and CPU times with those obtained using the integer programming approach. We also checked the algorithm's power to solve bigger instances. Finally, we present a real-like example to show that the algorithm correctly solves the incidents required by the company in a reasonable time.

Acknowledgements

This work has been supported by the ERDF; the MINECO/AEI grants MTM2014-53395-C3-1-P, MTM2017-87197-C3-1-P and ITC-20151247; and by the Xunta de Galicia (Grupos de Referencia Competitiva ED431C-2016-015 and Centro Singular de Investigación de Galicia ED431G/01). The authors would like to thank two anonymous referees for their very helpful suggestions to improve this article.

References

[1] C. Fikar, P. Hirsch, Home health care routing and scheduling: A review, Computers & Operations Research 77 (2017) 86–95.

[2] D. C. Paraskevopoulos, G. Laporte, P. P. Repoussis, C. D. Tarantilis, Resource constrained routing and scheduling: Review and research prospects, European Journal of Operational Research 263 (2017) 737–754.

[3] R. B. Bachouch, A. Guinet, S. Hajri-Gabouj, A decision-making tool for home health care nurses planning, Supply Chain Forum: an International Journal 12 (2011) 14–20.
[4] P. Cappanera, M. G. Scutellà, Joint assignment, scheduling, and routing models to home care optimization: A pattern-based approach, Transportation Science 49 (2014) 830–852.
[5] G. Carello, E. Lanzarone, A cardinality-constrained robust model for the assignment problem in home care services, European Journal of Operational Research 236 (2014) 748–762.
[6] K. D. Rest, P. Hirsch, Daily scheduling of home health care services using time-dependent public transport, Flexible Services and Manufacturing Journal 28 (2016) 495–525.
[7] P. M. Duque, M. Castro, K. Sörensen, P. Goos, Home care service planning. The case of Landelijke Thuiszorg, European Journal of Operational Research 243 (2015) 292–301.
[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 Journal of Operational Research 248 (2016) 428–443.
[9] P. Eveborn, P. Flisberg, M. Rönnqvist, Laps carean operational system for staff planning of home care, European Journal of Operational Research 171 (2006) 962–976.
[10] M. Misir, K. Verbeeck, P. De Causmaecker, G. V. Berghe, Hyper-heuristics with a dynamic heuristic set for the home care scheduling problem, in: Evolutionary Computation (CEC), 2010 IEEE Congress on, IEEE, 2010, pp. 1–8.
[11] S. Nickel, M. Schröder, J. Steeg, Mid-term and short-term planning support for home health care services, European Journal of Operational Research 219 (2012) 574–587.
[12] M. S. Rasmussen, T. Justesen, A. Dohn, J. Larsen, The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies, European Journal of Operational Research 219 (2012) 598–610.
[13] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.