CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

A Set Partitioning Heuristic for the Home Health Care Routing and Scheduling Problem

Florian Grenouilleau
Antoine Legrain
Nadia Lahrichi
Louis-Martin Rousseau

November 2017

CIRRELT-2017-70

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

A Set Partitioning Heuristic for the Home Health Care Routing and Scheduling Problem

Florian Grenouilleau[1,*], Antoine Legrain[2], Nadia Lahrichi[1], Louis-Martin Rousseau[1]

1. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Mathematics and Industrial Engineering, Polytechnique Montréal, P.O. Box 6079, Station Centre-Ville, Montréal, Canada H3C 3A7
2. Department of Industrial and Operations Engineering, University of Michigan, 1891 IOE Building, 1205 Beal Avenue, Ann Arbor, MI 48109-2117, USA

Abstract. The home health care routing and scheduling problem comprises the assignment and routing of a set of home care visits over the duration of a week. These services allow patients to remain in their own homes, thereby reducing governmental costs by decentralizing the care. In this work, we present a set partitioning heuristic which takes into account most of the industry's practical constraints. The developed method is based on a set partitioning formulation and a large neighborhood search framework. The algorithm solves a linear relaxation of a set partitioning model using the columns generated by the large neighborhood search. A constructive heuristic is then called to build an integer solution. Based on real instances provided by our industrial partner, the proposed method is able to provide a reduction in travel time by 37% and an increase by more than 16% in the continuity of care.

Keywords: OR in health services, routing, scheduling, set partitioning, large neighborhood search.

Acknowledgements. We thank our industrial partner Alayacare for their precious collaboration and financial support together with the grant agencies MITACS and MEDTEQ

---

* Corresponding author: florian.grenouilleau@cirrelt.net

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2017

1 Introduction

Home health care services improve patients' quality of life by helping them remain independent and in their own homes, often surrounded by family and friends, while maintaining their regular habits. From a governmental point of view, home care services decrease hospital congestion by freeing up hospital beds, which also results in reducing costs for these institutions [Macintyre et al., 2002].

In 2012, in Canada, more than 2.2 million people received home care services [Maire and Amanda, 2014]. These services are various: from personal support (bathing, dressing, housekeeping) to more specific tasks such as insulin injection or wound care. Due to the variety of tasks required, different medical specialties and skills are needed (e.g., personal social worker or nurse).

In this paper, we investigate the home health care routing and scheduling problem (HHCRSP) within a Canadian context. The problem is in determining the assignment of a set of home visits to a set of caregivers over the course of a week and the routing of these caregivers' workdays. The HHCRSP can be described as a multi-depot vehicle routing problem with time windows and time-dependent travel issues. Moreover, the home care context adds constraints focusing on the caregivers' skills and the patients' requirements (both mandatory and optional), as well as the management of the caregivers' work time contracts. Finally, the HHCRSP has a major concern which is the continuity of care, corresponding to the upkeep of a strong patient-caregiver relationship. The work presented here has been done in collaboration with a Montreal start-up, Alayacare, which has developed an operations management platform for Canadian home health care agencies. They aim to provide their clients a flexible optimization module which solves real-life instances with minimal computational time constraints (no more than 10 minutes).

From our knowledge, the HHCRSP is a 20-year-old problem [Begur et al., 1997, Cheng and Rich, 1998] that was originally solved over a daily planning horizon. The problem, thereafter, has been extended to a weekly horizon that allows for better coping with the reality of some constraints, such as the patients' care plan and/or the continuity of care. Some methods using branch-and-price [Gamst and Jensen, 2012], branch-and-price-and-cut algorithm [Trautsamwieser and Hirsch, 2014] or integer linear based method [Borsani et al., 2006, Torres-Ramos et al., 2014] have been proposed, but the complexity of the problem leads to scalability issues. To cope with these issues, methods based on heuristics or meta-heuristics have been developed using frameworks such as swarm optimization [Akjiratikarl et al., 2007], large neighborhood search [Di Gaspero and Urli, 2014] or harmony search [Lin et al., 2017]. In Nickel et al. [2012], the problem is split in two: the *master problem*, which uses a constructive heuristic and an ALNS to build a feasible assignment of the visits, and the *operational problem*, which integrates the last minute changes (e.g., visit cancellation or sick caregiver) into the current schedule with an insertion heuristic and a tabu-search. Finally, Duque et al. [2015] propose a two-phase method based on a set partitioning formulation. The first phase produces pools of visit patterns and solves the patterns' assignment using Cplex. Then, the second phase improves the best patterns' assignment with a local search procedure that swaps patients' visits to reduce travel time and maximize patients and nurses' preferences. For more references, we refer the reader to two excellent surveys published recently [Cissé et al., 2017, Fikar and Hirsch, 2017].

In this work, we present a set partitioning heuristic (SPH). This method is based on the heuristic concentration principle [Rosing and ReVelle, 1997]. The goal of our SPH is to solve a set partitioning formulation of the HHCRSP using the columns (feasible routes) generated by a Large Neighborhood Search (LNS) [Shaw, 1998]. Due to the necessity to produce high quality solutions in a small computational time, the SPH solves a linear relaxation of the set partitioning formulation and a constructive heuristic is then applied to build an integer solution based on the solution found.

This paper presents three major contributions. First, the proposed method takes into account a large set of practical constraints and solves instances covering up to 430 visits in less than 10 minutes. Second, we propose an improved heuristic concentration approach allowing for the quality of an exact method with the rapidity of a heuristic. Finally, we propose new LNS' operators, specifically designed for the HHCRSP, which permit the extension of the search space to find new and improved solutions.

The paper is organized as follows. Section 2 presents the problem and its formulation. Section 3 details our approach and Section 4 shows the computational results on generated and real instances. Finally, a conclusion of the study is drawn in Section 5.

2 Problem definition

The home health care routing and scheduling problem can be described as a multi-attribute vehicle routing problem. We define the sets P of patients and C of caregivers. The objective is, while minimizing the sum of the penalties, to determine the caregivers' routes over the horizon of H days ($H = 7$ in our context) in order to visit each patient a required number of times. The caregivers' assignments must take into account patients' mandatory and optional requirements, caregivers' skills, and forbidden assignments (e.g., due to some allergies or personal conflict). The routing part of the problem must cope with patients' availability (days and time windows) and caregivers' work shifts. Caregivers' work contracts (i.e., minimum and maximum amount of working time per day and week) have to be managed as well. Finally, the impact of traffic delays on travel time are taken into account, through a time-dependent distance matrix.

For each patient $p \in P$, we define a number n_p of required visits of duration dur_p, a subset $D_p \subseteq [1, ..., H]$ of available days and a hard time-window $[e_p^d, l_p^d]$ for each available day $d \in D_p$. Moreover, we also define two lists M_p and O_p that respectively contain the mandatory and optional expertise required by the assigned caregiver. The optional expertises could be described as patient's preferences about, for example, the gender or the language spoken by the assigned caregiver. Finally a set $\overline{C}_p$ of forbidden caregivers is attached to each patient.

For each caregiver $c \in C$, we similarly define a list E_c of expertise, a soft minimum $\underline{w}_c^w$ and maximum $\overline{w}_c^w$, work times over the week, and a subset $D_c \subseteq [1, ..., H]$ of workdays. Each of these workdays d also has a time-window $[a_c^d, b_c^d]$ and a soft minimum $\underline{w}_c^d$ and maximum $\overline{w}_c^d$ of work times.

Every patient and caregiver have their home location (respectively l_p and l_c) that belongs to a set L of possible zip codes. Finally, the continuity of care measures the strength of a patient-caregiver relationship with a score $CC_{p,c}$ which is equal to the number of times the caregiver c has visited the patient p the previous week.

We propose to formulate the HHCRSP as a set partitioning problem (SPP) that aims at selecting the best routes for each caregiver among a set Ω of daily feasible caregivers' routes. Each route $\omega \in \Omega$ takes into account the patients' mandatory requirements, the forbidden caregivers, the caregivers' skills, the time-windows, and the time-dependent travel times.

Each route is assigned a length len_ω, travel time tt_ω and number of missing optional expertises $\overline{o}_\omega$ for the visited patients. We also define the subsets $\Omega_d \subset \Omega$ and $\Omega_c \subset \Omega$ that correspond to the routes associated respectively to day d and caregiver c. The cost c_ω of each route $\omega \in \Omega$ is defined as a weighted sum of soft constraints' penalties :

$c_\omega = \gamma_1.\overline{o}_\omega + \gamma_2.tt_\omega + \gamma_3.\sum_{p \in P} a_{\omega,p}.f_1(CC_{p,c}) + \gamma_4.f_2(len_\omega)$, where $a_{\omega,p}$ equals to 1 if the route ω visits patient p.

The first term of the cost function corresponds to the missing optional expertises penalty. The second term corresponds to the travel time cost and the third, to the continuity of care penalty. Finally, the last term is the penalty corresponding to the non-respect of the minimum or maximum daily work time for each caregiver.

The penalty function f_1 is given by :

$$f_1(CC_{p,c}) = \begin{cases} 1 & \text{if } CC_{p,c} = 0 \\ \frac{2}{3} & \text{if } 1 \le CC_{p,c} \le 2 \\ \frac{1}{3} & \text{otherwise} \end{cases}$$

Finally, the work time penalty function f_2 is described as follows:

$$f_2(len_\omega) = \begin{cases} \underline{w}_d - len_\omega & \text{if } len_\omega < \underline{w}_d \\ len_\omega - \overline{w}_d & \text{if } len_\omega > \underline{w}_d \\ 0 & \text{otherwise} \end{cases}$$

The decision variables of the problem are given by :

- x_ω which equals 1 if the route ω is selected, 0 otherwise;
- o_c which measures the weekly overtime for caregiver c;
- u_c which also measures the weekly idle time for caregiver c;
- z_p which counts the number of unscheduled visits for patient p.

The corresponding SPP formulation is defined as follows:

$$(SPP) : \min \sum_{\omega \in \Omega} c_\omega x_\omega + \beta_1 . \sum_{c \in C} (o_c + u_c) + \beta_2 . \sum_{p \in P} z_p \quad (1)$$

$$\text{subject to: } \sum_{\omega \in \Omega_d} a_{\omega,p} x_\omega \le 1 \quad \forall p \in P, d \in D_p \quad (2)$$

$$\sum_{\omega \in \Omega} a_{\omega,p} x_\omega + z_p = n_p \quad \forall p \in P \quad (3)$$

$$\sum_{\omega \in \Omega_d \cap \Omega_c} x_\omega \le 1 \quad \forall c \in C, d \in D_c \quad (4)$$

$$\sum_{\omega \in \Omega} l_\omega x_\omega + u_c \ge \underline{w}_c^w \quad \forall c \in C \quad (5)$$

$$\sum_{\omega \in \Omega} l_\omega x_\omega - o_c \le \overline{w}_c^w \quad \forall c \in C \quad (6)$$

$$x_\omega \in \{0, 1\} \quad \forall \omega \in \Omega \quad (7)$$

$$z_p \ge 0 \quad \forall p \in P \quad (8)$$

$$o_c, u_c \ge 0 \quad \forall c \in C \quad (9)$$

The objective function (1) corresponds to a weighted sum of costs associated, respectively, to the routes, the weekly caregivers' overtimes and idle time and the unscheduled visits. Constraints (2) ensure that patient p is visited a maximum of once per day, Constraints (3) count the number of unscheduled visits per patient. Then, Constraints (4) ensure that no more than one route per day is assigned to each caregiver. Finally, Constraints (5) – (6) measure, respectively, the weekly idle time and overtime. The domains of the variables are defined by Constraints (7) – (9).

3 Resolution Method

In this section, we present the set partitioning heuristic (SPH). The proposed SPH is a matheuristic based on the resolution of the SPP presented in the section 2. This method is based on the heuristic concentration principle [Rosing and ReVelle, 1997]. The aim of the heuristic concentration is to keep the best solutions found by a heuristic procedure and then use a set partitioning that combines parts of these solutions to create a better one. This combination of heuristic and exact approaches have already been used for the VRPTW [Muter et al., 2010, Mendoza et al., 2016]. In our method, the possible SPP's routes are found using a Large Neighborhood Search (LNS). The LNS [Shaw, 1998] is a meta-heuristic using the $ruin - and - recreate$ principle [Schrimpf et al., 2000]. This method, starting from an initial solution, iteratively destroys a part of the current solution, then repairs it to improve its quality. The current and best solutions are then updated if necessary. A full description of the LNS can be found in Gendreau and Potvin [2010].

Due to the computational time required to solve the SPP with a great number of routes, the SPH solves the relaxation ($Relaxed_{SPP}$) of this model by relaxing the integrity of the x_ω decision variables ($\bar{x}_\omega$). Then, a constructive heuristic ($Heur_{SPP}$) is applied and new integer solution are built based on the $Relaxed_{SPP}$'s result.

An overview of the SPH is given by the Algorithm 1. The first part of the algorithm is based on the LNS' procedure described earlier (initial solution, destruction, repair, analysis). Then, at the end of each segment (i.e., a block of N iterations), a sub-procedure is called. This procedure solves the $Relaxed_{SPP}$ and applies the $Heur_{SPP}$.

```
Find an initial solution ;
while No termination criteria met do
    s ← currentSolution ;
    Select and apply a destroy operator on s ;
    Select and apply a repair operator on s ;
    Analyze the solution s ;
    if A end of segment is met then
        Solve Relaxed_SPP ;
        Apply Heur_SPP ;
    end
end
Return the best found solution ;
```

Algorithm 1: SPH

3.1 Implementation details

Initial solution We use a greedy heuristic to build the initial solution. We first sort the visits in decreasing order of their durations, then, following this order, we insert each visit at the lowest-cost position. The unscheduled visits are stored in a list and stay there until the first repair procedure.

Classic operators For the LNS' iterations, a part of the used operators are classic ones such as $WorstRemoval$, $RandomRemoval$, $Greedy\ Heuristic$, $regret$-2 and $regret$-3 from Ropke and Pisinger [2006] and $RelatedRemoval$ from Shaw [1998]. The five new operators are described in the subsection 3.3.

Range of destruction The number q of visits destroyed at each iteration is randomly drawn in a range [$min\_percent$,$max\_percent$]*$Sched_s$ where $Sched_s$ is the number of visits scheduled in the impacted solution s.

Solution Analysis After the destroy and repair procedures, the created solution is analyzed to decide if its quality is good enough to be kept as a best or current solution. Three cases may occur in this context :

1. the new solution is better than the best found, the LNS updates the best and current solutions with the new one;
2. the new solution is better than the current solution, only the current solution is updated;
3. the new solution is worse than the current solution, a simulated annealing accept criterion is then used to either accept or refuse it.

This simulated annealing accept criterion [Kirkpatrick et al., 1983] accepts the new solution with a probability $e^{-\frac{f(s_{new})-f(s_{cur})}{T}}$ where $f(s_{new})$ and $f(s_{cur})$ are respectively the value of the new and current solutions. The value T is the current temperature of the problem which decreases at each simulated annealing call, according to the relation subscript $T_{n+1} = T_n \times c$ where $0 < c < 1$ is the decrease coefficient. According to Ropke and Pisinger [2006], the decrease coefficient c and the initial temperature T_0 are respectively to 0.99975 and $1.05 \times f(s_0)$, where s_0 is the initial solution.

Termination criterion The SPH ends when reaching either a maximum number of LNS' iterations or a maximum computational time.

Management of the time-dependent travel time The LNS implements a dynamic computation of the time-dependent travel times which is based on the algorithm described by Ichoua et al. [2003]. This algorithm respects the FIFO logic and computes the travel times according to the start and end locations and the departure time.

3.2 Constructive Heuristic ($Heur_{SPP}$)

After the resolution of the SPP's relaxation ($Relaxed_{SPP}$), the $Heur_{SPP}$ procedure is called to build an integer solution according to the resultant relaxed values $\bar{x}_\omega$. An overview of the method is given by the algorithm 2.

```
Create the list L, copy of the routes in Ω, sorted in decreasing order of the values x̄_ω from the last
  Relaxed_SPP
Empty solution s
forall route ω in L do
  | forall patient visit v in ω's visit list do
  |   | if The patient of the visit v has all his/her visits scheduled in s then
  |   |   | remove v from ω's visit list
  | end
  | if ω's visit list is not empty then
  |   | Reschedule ω with the remaining visits
  |   | Insert the route ω in s
end
if The solution s is better than the best found solution then
  | Update the best found solution with s
end
```

Algorithm 2: $Heur_{SPP}$

3.3 New LNS operators

In order to focus the search on some difficult aspects of the problem, some problem-specific destroy and repair operators have been implemented in the LNS.

New destroy operators Let us recall that q, the number of destroyed visits, is randomly selected at each LNS' iteration. The developed destroy operators are as follows :

1. The *ServiceRemoval* operator randomly selects a patient and removes all his/her scheduled visits. This process is repeated until at least q visits are removed. This new operator permits a reset of the assigned visit days of the patient and potentially creates a new pattern of visits during the repair part.
2. The *FlexibleAvailRemoval* operator deletes from the current schedule the patients with the highest flexibility (i.e., highest value of $\frac{|D_p|}{n_p}$). Iteratively, the most flexible patient is selected and all its scheduled visits are removed from the current schedule. The patients list is scanned this way until q visits are removed.
3. The *DualRemoval* operator uses the dual values from the last $Relaxed_{SPP}$ resolution. Based on constraints (3), this operator sorts the patients in non-decreasing order of their dual values, then iteratively selects the patient at the top of the list (lowest dual value), and removes his/her visits. The process is repeated until q visits are removed like the other destroy operators.

New repair operators For the proposed LNS, two new repair operators have been created :

1. The *RandomService* operator randomly chooses one of the patients for which some visits are not scheduled. A lowest-cost insertion logic is used to schedule his/her visits over the horizon. This process is repeated until every patient with missing visits has been tested.
2. The *DualRepair* operator focuses on the patients with the highest dual values. It sorts the patient in decreasing order of their dual values, based on constraints (3) of the last $Relaxed_{SPP}$'s resolution. Then, the operator follows this ordered list and tries to schedule as many visits as possible for each patient using again a lowest-cost insertion logic.

Due to the fact that the dual values come from the $Relaxed_{SPP}$, the dual operators (*DualRemoval*, *DualRepair*) can't be used in the first LNS's segment (first N iterations). They are introduced in the operators lists at the end of the first $Relaxed_{SPP}$.

4 Computational Results

This section presents some computational experiments: first, we compare our SPH with a classic LNS approach on generated instances; then, we analyze the improvements permitted by our SPH on real instances provided by our industrial partner. The proposed algorithm has been implemented in C++ and all the tests are run on a Linux 3.07Gh computer with 20G Ram CPU. The termination criterion are set to 10 minutes and 10^6 LNS' iterations. The *min_percent* and *max_percent* have been respectively set to 2% and 5% and the size of a segment is 10^3 iterations. Finally, the penalties' weights (γ_1,γ_2,γ_3,γ_4,β_1,β_2,) have been fixed after preliminary evaluations in collaboration with Alayacare.

4.1 Experiments on generated-instances

In order to test the proposed SPH, we have based our analysis on a benchmark of 60 instances : three sets (Small, Medium, Large) of 20 instances corresponding to the different problem's sizes that must be solved by the algorithm. An overview of the instances' characteristics is given in the table 1.

Instance	Patient	Visits	Caregiver	Workdays
Small	40	120	5	25
Medium	80	225	10	45
Large	150	430	20	90

Table 1: Characteristics of the generated instances

These sets have been randomly generated based on real instances' characteristics provided by our industrial partner and each value has a predefined range. The instances' generation is based on 5 different expertise, 141 possible locations, and several parameters described in Tables 2 and 3.

Parameter	Name	Minimum	Maximum
n_p	Number of visits	1	7
dur_p	Duration of visits	40	60
$\|M_p\|$	Mandatory expertise	1	2
$\|O_p\|$	Optional expertise	0	2
$\frac{l_p^d - e_p^d}{dur_p}$	Time-window's size	2	4

Table 2: Services' parameters for the generated instances

Parameter	Name	Minimum	Maximum
$\underline{w}_c^w$	Minimum week work time	0 min	600 min
$\overline{w}_c^w$	Maximum week work time	1200 min	2400 min
$b_c^d - a_c^d$	time-window's size	420 min	720 min
$\underline{w}_c^d$	Minimum day work time	0	$\frac{30}{100}(b_c^d - a_c^d)$
$\overline{w}_c^d$	Maximum day work time	$\frac{80}{100}(b_c^d - a_c^d)$	$\frac{100}{100}(b_c^d - a_c^d)$
$\|E_c\|$	Expertise list	2	3

Table 3: Employees' parameters for the generated instances

In order to observe the impact of the proposed operators, we define 2 groups of operators :

- CL : The classic operators with $WorstRemoval$, $RandomRemoval$, $RelatedRemoval$ for the destroy part and *Greedy Heuristic*, *regret-2* and *regret-3* for the repair ones.
- NW : The new operators : $ServiceRemoval$, $FlexibleAvailRemoval$ and $DualRemoval$ for the destroy operators, $RandomService$ and $DualRepair$ for the repair ones. These operators necessitate the resolution of the $Relaxed_{SPP}$.

Moreover, to test the impact of the $Heur_{SPP}$, we distinguish the use or not of this algorithm.

For this analysis, 10 runs of each instance have been computed for three different scenarios (CL, CL + NW and CL + NW + $Heur_{SPP}$). The presented results are based on the average of the best found solutions' costs over the 10 runs. The figures 1, 2 and 3 present the comparison of the three scenarios. The values correspond to the gap between each scenario's value and the value of the CL one. According to these results, we can observe that, on average, the new operators ($CL + NW$ scenario), by extending the search space, find better solutions and reduce the solutions' cost for the small, medium and large instances by respectively 7.63%, 10.06% and 2.34% (see tables 5, 6 and 7 in Appendix **??**). The reduced improvements produced by the new operators on the large instances could be due to the reduced number of iterations done (see table 8 in Appendix **??**). This reduction of the number of iteration (32211 for CL to 24101 for $CL + NW$) is probably caused by the time spent in the resolution of $Relaxed_{SPP}$ at each end of segment.

Furthermore, we can observe that the $Heur_{SPP}$ ($CL + NW + Heur_{SPP}$ scenario) is able to find the best solutions for all instances : the improvements for the three instances' sets are respectively 13.76%, 20.82% and 14.39%. According to these observations, we'll keep the $CL + NW + Heur_{SPP}$ scenario for the real instances' resolution.

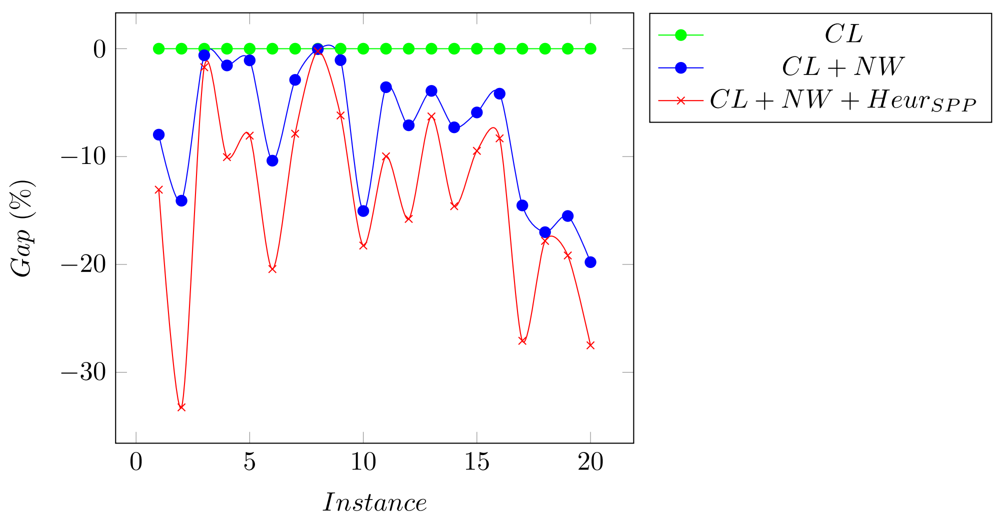

Figure 1: Comparison of the cost for the small instances

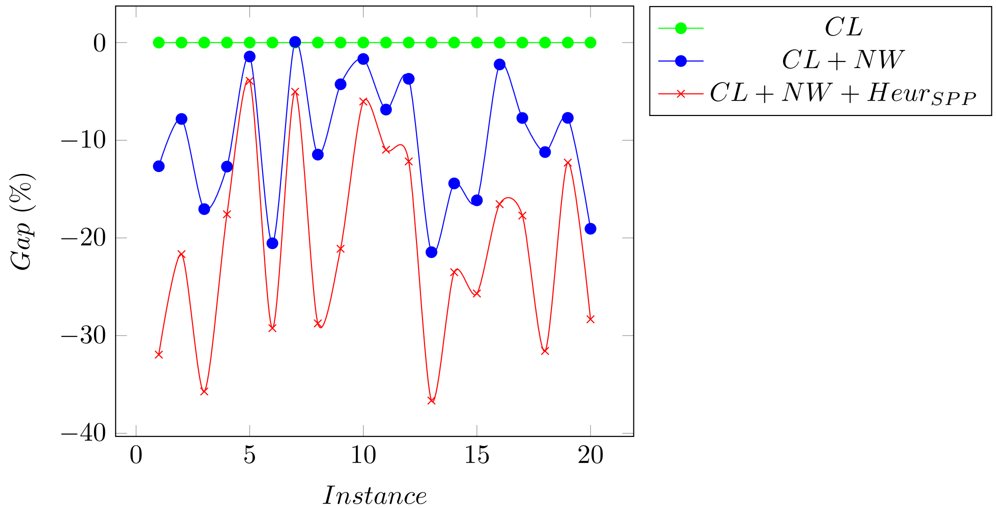

Figure 2: Comparison of the cost for the medium instances

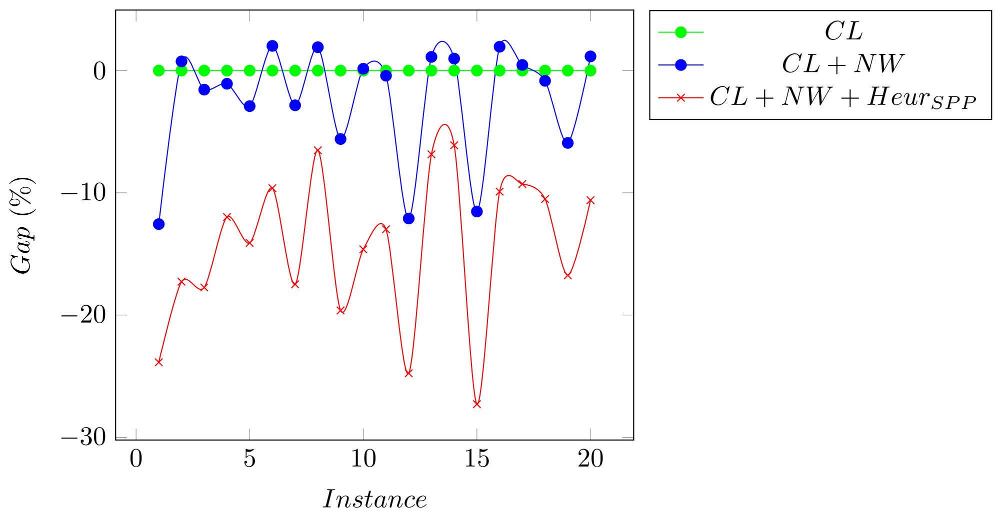

Figure 3: Comparison of the cost for the large instances

4.2 Real-World Instances

In this section, we describe the tests performed on instances from an Alayacare's client. For the studied client, the objective was to analyze the improvements both in terms of travel time and continuity of care provided by the proposed method.

In these experiments, 4 instances representing 4 different weeks have been used. These instances are described as $P\_V\_C\_R$ where P is the number of patients, V the number of visits, C the number of caregivers and R the number of routes (number of workdays). For these instances, the chosen patients were homogeneous, so the same expertise were needed. The available days correspond to actual patients' visits' days (i.e. $|D_p| = n_p$ for each patient). The patients' time windows were designed around their actual visit times. For the employees, their workdays, work time contracts and time windows were given by the client.

The figure 4 presents the distribution of the number of visits per patient for the real instances. According to this figure, the majority of patients only need 1 or 2 visits per week.

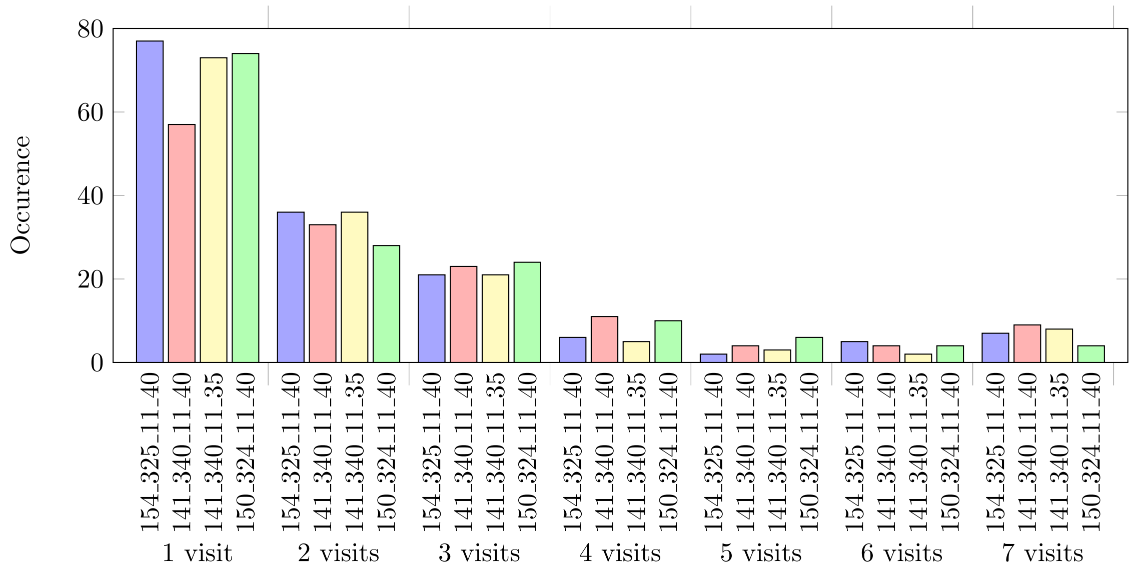

Figure 4: Distribution of the number of visit per patient

A comparison of Alayacare's current client solutions and our SPH's solutions on these 4 instances is presented in Table 4. According to the client's will, we focus here on two major indicators, the total travel time (TT) and the continuity of care (CC, i.e., the percentage of scheduled visits for which the patient p and the caregiver c have $CC_{p,c} \neq 0$).

	Current solution		SPH**'s solution**		Δ	
Instance	TT	CC	TT	CC	TT	CC
154_325_11_40	4361.16	60%	**2431.62**	**75.94%**	-44.24%	+15.94%
141_340_11_40	4549.03	62.33%	**2833.18**	**79.05%**	-37.72%	+16.72%
148_311_11_35	3832.94	71.69%	**2571.29**	**85.98 %**	-32.92%	+14.29%
150_324_11_40	3686.57	64.43%	**2464.22**	**82.10%**	-33.16%	+17.67%
Mean	4107.43	64.61%	2575.08	80.77%	**-37.01%**	**+16.16%**

Table 4: Comparison of the actual solutions with those produced by our approach

According to the Table 4, our approach improves the solutions both in terms of travel time and continuity of care. On average, the proposed algorithm reduces the total travel time by 37.01% and increases the continuity of care by 16.16%. These results show that the use of such method by Alayacare's clients could lead to large improvement in term of costs reduction and quality of service.

5 Conclusion

The HHCRSP is a complex problem due to the simultaneous management of the assignment (requirements, skills, continuity of care, forbidden assignments) and routing (travel time, work time contracts, impact of the traffic) constraints. Nevertheless, we have proposed a set partitioning heuristic able to cope with all these requirements.

The presented method is firstly based on a set partitioning formulation of the problem. The resolution of this set partitioning is done in two phases : the resolution of the relaxation ($Relaxed_{SPP}$) followed by a constructive heuristic ($Heur_{SPP}$). To populate the SPP's columns, we developed a LNS procedure. This LNS has three benefits, it allows us : to generate possible routes for the SPP, to always have a feasible primal solution and, during the segments, to continuously improve the best found solution. To extend the LNS' search space, five new operators have also been proposed.

According to the results, we observed that the new operators and the constructive heuristic permit a dramatic reduction in term of solutions' costs for the generated instances (respectively 13.76%, 20.82% and 14.39% for the small, medium and large sets). On the real instances, the algorithm permitted, on average, a 37% reduction in travel time and a 16% increase in the continuity of care. The developed method has been approved by our industrial partner and integrated in their software. It's used by Alayacare's clients since November 2017.

References

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

S. V. Begur, D. M. Miller, and J. R. Weaver. An integrated spatial dss for scheduling and routing home-health-care nurses. *Interfaces*, 27(4):35–48, 1997.

V. Borsani, A. Matta, G. Beschi, and F. Sommaruga. A home care scheduling model for human resources. In *Service systems and service management, 2006 international conference on*, volume 1, pages 449–454. IEEE, 2006.

E. Cheng and J. L. Rich. A home health care routing and scheduling problem. *Houston, Texas*, 1998.

M. Cissé, S. Yalçındağ, Y. Kergosien, E. Şahin, C. Lenté, and A. Matta. Or problems related to home health care: A review of relevant routing and scheduling problems. *Operations Research for Health Care*, 2017.

L. Di Gaspero and T. Urli. A cp/lns approach for multi-day homecare scheduling problems. In *International Workshop on Hybrid Metaheuristics*, pages 1–15. Springer, 2014.

P. M. Duque, M. Castro, K. Sörensen, and P. Goos. Home care service planning. the case of landelijke thuiszorg. *European Journal of Operational Research*, 243(1):292–301, 2015.

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

M. Gamst and T. S. Jensen. A branch-and-price algorithm for the long-term home care scheduling problem. pages 483–488, 2012.

M. Gendreau and J.-Y. Potvin. *Handbook of metaheuristics*, volume 2. Springer, 2010.

S. Ichoua, M. Gendreau, and J.-Y. Potvin. Vehicle dispatching with time-dependent travel times. *European journal of operational research*, 144(2):379–396, 2003.

S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, et al. Optimization by simulated annealing. *science*, 220(4598): 671–680, 1983.

C.-C. Lin, L.-P. Hung, W.-Y. Liu, and M.-C. Tsai. Jointly rostering, routing, and rerostering for home health care services: A harmony search approach with genetic, saturation, inheritance, and immigrant schemes. *Computers & Industrial Engineering*, 2017.

C. R. Macintyre, D. Ruth, and Z. Ansari. Hospital in the home is cost saving for appropriately selected patients: a comparison with in-hospital care. *International Journal for Quality in Health Care*, 14(4): 285–293, 2002.

S. Maire and B. Amanda. *Receiving care at home*. Statistics Canada, 2014.

J. E. Mendoza, L.-M. Rousseau, and J. G. Villegas. A hybrid metaheuristic for the vehicle routing problem with stochastic demand and duration constraints. *Journal of Heuristics*, 22(4):539–566, 2016.

İ. Muter, Ş. İ. Birbil, and G. Şahin. Combination of metaheuristic and exact algorithms for solving set covering-type optimization problems. *INFORMS Journal on Computing*, 22(4):603–619, 2010.

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

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

K. Rosing and C. ReVelle. Heuristic concentration: Two stage solution construction. *European Journal of Operational Research*, 97(1):75–86, 1997.

G. Schrimpf, J. Schneider, H. Stamm-Wilbrandt, and G. Dueck. Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics*, 159(2):139–171, 2000.

P. Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In *International Conference on Principles and Practice of Constraint Programming*, pages 417–431. Springer, 1998.

A. Torres-Ramos, E. Alfonso-Lizarazo, L. Reyes-Rubiano, and C. Quintero-Araújo. Mathematical model for the home health care routing and scheduling problem with multiple treatments and time windows. In *Proceedings of the 1st International Conference on Mathematical Methods & Computational Techniques in Science & Engineering*, pages 140–145, 2014.

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

	CL	CL + NW		CL + NW + $Heur_{SPP}$	
	Value	Value	Gap	Value	Gap
Small_01	753650.3069	693571.5086	**-7.97%**	655170.3155	**-13.07%**
Small_02	388578.3983	333827.2444	**-14.09%**	259383.9622	**-33.25%**
Small_03	261321.7531	259734.4134	**-0.61%**	256827.8812	**-1.72%**
Small_04	307137.0779	302363.2317	**-1.55%**	276264.5791	**-10.05%**
Small_05	324180.2087	320682.1948	**-1.08%**	298047.6945	**-8.06%**
Small_06	602088.1618	539567.8536	**-10.38%**	479109.7101	**-20.43%**
Small_07	273176.1163	265288.0998	**-2.89%**	251640.4849	**-7.88%**
Small_08	1528431.827	1527824.961	**-0.04%**	1524747.939	**-0.24%**
Small_09	394693.7361	390526.2682	**-1.06%**	370272.7138	**-6.19%**
Small_10	469118.0827	398505.7739	**-15.05%**	383520.661	**-18.25%**
Small_11	283330.1056	273207.2517	**-3.57%**	255041.2672	**-9.98%**
Small_12	938840.888	872176.9512	**-7.10%**	790714.4061	**-15.78%**
Small_13	244551.8327	234983.1013	**-3.91%**	229196.1026	**-6.28%**
Small_14	860348.6968	797670.6448	**-7.29%**	734774.0971	**-14.60%**
Small_15	993613.4881	934921.4549	**-5.91%**	899489.8394	**-9.47%**
Small_16	447580.6052	428907.5975	**-4.17%**	410401.4716	**-8.31%**
Small_17	1096303.373	937055.3012	**-14.53%**	799396.3975	**-27.08%**
Small_18	559169.9291	464071.3918	**-17.01%**	459508.1039	**-17.82%**
Small_19	521073.6217	440239.0151	**-15.51%**	421205.0284	**-19.17%**
Small_20	881554.9213	715901.5364	**-18.79%**	639114.1441	**-27.50%**
Mean Gap			**-7.63%**		**-13.76%**

Table 5: Comparison of the scenarios for the small instances

	CL	CL + NW		CL + NW + $Heur_{SPP}$	
	Value	Value	Gap	Value	Gap
Medium_01	1588963.974	1387766.741	**-12.66%**	1081305.183	**-31.95%**
Medium_02	828816.0153	764033.1878	**-7.82%**	649277.7471	**-21.66%**
Medium_03	1286191.22	1066939.709	**-17.05%**	826778.7154	**-35.72%**
Medium_04	800638.5202	698931.1735	**-12.70%**	659917.3411	**-17.58%**
Medium_05	618752.8272	609918.1006	**-1.43%**	594481.4027	**-3.92%**
Medium_06	887273.5803	704931.9291	**-20.55%**	627804.6367	**-29.24%**
Medium_07	888716.3092	889382.7083	0.07%	844053.2943	**-5.03%**
Medium_08	785631.1344	695492.8756	**-11.47%**	559769.3352	**-28.75%**
Medium_09	685023.2233	655755.5022	**-4.27%**	540411.6994	**-21.11%**
Medium_10	786320.4112	773141.127	**-1.68%**	738833.9906	**-6.04%**
Medium_11	937630.5887	873310.6694	**-6.86%**	834727.7175	**-10.97%**
Medium_12	596877.7024	574721.1511	**-3.71%**	524259.2039	**-12.17%**
Medium_13	1039973.258	816774.9148	**-21.46%**	658848.834	**-36.65%**
Medium_14	708509.2346	606313.0627	**-14.42%**	541961.2916	**-23.51%**
Medium_15	801160.0585	671760.9483	**-16.15%**	595359.6597	**-25.69%**
Medium_16	845822.0983	826898.2096	**-2.24%**	705922.754	**-16.54%**
Medium_17	776339.8731	716435.7644	**-7.72%**	638842.8314	**-17.71%**
Medium_18	2207257.988	1933365.621	**-12.41%**	1510478.666	**-31.57%**
Medium_19	607654.9893	560813.015	**-7.71%**	532946.6944	**-12.29%**
Medium_20	844354.0591	683449.5749	**-19.06%**	605117.078	**-28.33%**
Mean Gap			**-10.06%**		**-20.82%**

Table 6: Comparison of the scenarios for the medium instances

	CL	CL + NW		CL + NW + $Heur_{SPP}$	
	Value	Value	Gap	Value	Gap
Large_01	1315840.266	1150528.851	**-12.56%**	1001922.789	**-23.86%**
Large_02	1271025.392	1280565.368	0.75%	1051504.59	**-17.27%**
Large_03	1275166.168	1255270.505	**-1.56%**	1048994.785	**-17.74%**
Large_04	1349729.927	1335133.742	**-1.08%**	1188032.835	**-11.98%**
Large_05	1252057.507	1215604.269	**-2.91%**	1075480.953	**-14.10%**
Large_06	1163047.195	1186513.277	2.02%	1051132.77	**-9.62%**
Large_07	1171658.516	1138382.318	**-2.84%**	966833.2807	**-17.48%**
Large_08	1022707.503	1042276.777	1.91%	956056.2688	**-6.52%**
Large_09	1253375.629	1183201.189	**-5.60%**	1007451.183	**-19.62%**
Large_10	1128399.049	1130063.193	0.15%	963391.3816	**-14.62%**
Large_11	1249775.597	1244545.952	**-0.42%**	1087580.947	**-12.98%**
Large_12	1270174.657	1116505.442	**-12.10%**	955662.294	**-24.76%**
Large_13	1058909.794	1070766.139	1.12%	986339.5384	**-6.85%**
Large_14	988281.5901	997946.7649	0.98%	927882.2825	**-6.11%**
Large_15	1545000.491	1366852.108	**-11.53%**	1123416.863	**-27.29%**
Large_16	1239669.083	1263805.393	1.95%	1116903.185	**-9.90%**
Large_17	1036061.191	1040816.364	0.46%	939885.2916	**-9.28%**
Large_18	1042017.091	1033234.667	**-0.84%**	932547.2436	**-10.51%**
Large_19	1250220.558	1176267.759	**-5.92%**	1040813.506	**-16.75%**
Large_20	1128135.796	1141223.339	1.16%	1008533.857	**-10.60%**
Mean Gap			**-2.34%**		**-14.39%**

Table 7: Comparison of the scenarios for the large instances

	CL		CL + NW		CL + NW + Heur	
	time (s)	Iterations	time (s)	Iterations	time (s)	Iterations
Small_Instances	149.8	100000	164.9	100000	163.8	100000
Medium_Instances	517.5	98607	597.6	79834	584	84928
Large_Instances	601.2	32211	602.2	24101	602.4	24258

Table 8: Comparison of the computation time and number of iteration for the three scenarios (Average over the 10 runs)