Annals of Operations Research (2023) 320:573–601
https://doi.org/10.1007/s10479-021-04222-w

ORIGINAL RESEARCH

Stochastic home health care routing and scheduling problem with multiple synchronized services

Mohammed Bazirha[1] · Abdeslam Kadrani[1] · Rachid Benmansour[1,2]

Accepted: 28 July 2021 / Published online: 9 August 2021

Abstract

The home health care (HHC) covers a wide range of health care services carried out in patients' home in case of illness, injury or aging. Each caregiver should as far as possible adhere to the schedule set by the decision maker. However, unforeseen events would sometimes occur and delay the delivery of care services, which will qualify the service as poor or even risky. Deterministic models ignore this uncertainty, which can arise at any time and will therefore lead to non-compliance with the predefined schedule. Furthermore, patients need several care activities per day, and some of them require to be simultaneous by their nature such as dressing, getting out of bed and bathing. In this work, a stochastic programming model with recourse (SPR model) is proposed to deal with the home health care routing and scheduling problem (HHCRSP) where uncertainties in terms of traveling and caring times that may occur as well as synchronization of services are considered. The objective is to minimize the transportation cost and the expected value of recourse, which is estimated using Monte Carlo simulation. The recourse is defined as a penalty cost for patients' delayed services and a remuneration for caregivers' extra working time. The deterministic model is solved by CPLEX, the genetic algorithm (GA) and the general variable neighborhood search (GVNS) based heuristics. The SPR model is solved by Monte Carlo simulation embedded into the GA. Computational results highlight the efficiency of GVNS and GA based heuristics and the complexity of the SPR model in terms of CPU running times.

Keywords Variable neighborhood search · Genetic algorithm · Home health care · Routing and scheduling · Multiple synchronized services · Stochastic programming model with recourse

✉ Mohammed Bazirha
mbazirha@insea.ac.ma

Abdeslam Kadrani
akadrani@insea.ac.ma

Rachid Benmansour
r.benmansour@insea.ac.ma

[1] SI2M Laboratory INSEA, Rabat, Morocco

[2] LAMIH UMR CNRS 8201, Valenciennes, France

1 Introduction

The HHC is an evolving research area that has received growing consideration in the recent years due to increasing life expectancy and low birth rates. According to the world report on ageing and health 2015 (WHO and Course 2017), individuals are expected on average to live to age 77; 15 of these years would be passed with some kind of disability. The percentage of elderly people is increasing in European countries and is anticipated to rise still further in the following years (Tarricone and Tsouros 2008). Furthermore, family members live more often isolated from each other since children move to other cities in order to find jobs (Mankowska et al. 2014) or to pursue studies while the parents stay in the hometown. The HHC will help elderly people to live autonomously for as long as possible even with an injury, illness or disabling disease. It covers a wide range of services that may involve medical care such as nursing, physical therapy and speech therapy. It may include helping elderly individuals with activities of daily living such as eating, dressing and bathing. It can also involve assistance with cooking, cleaning and other housekeeping. HHC companies receive several requests of services, so the decision maker has to assign qualified caregivers to patients and define their routes with respect to the available resources and the patient preferences in terms of the availability periods (preferred time window). This problem is well known as the home health care routing and scheduling problem (HHCRSP), which is a combination of two NP-hard problems: the vehicle routing problem with time windows (Bräysy and Gendreau 2005) and the nurse rostering problem (Burke et al. 2004).

Several models and methods have been proposed in the literature to deal with the HHCRSP but most of them are deterministic and generally less robust. The predefined schedule must be revised for any change in practical situations. Otherwise, there will probably be delays in the services for patients who have been not yet visited, which will cause their dissatisfaction. Travel and service times are critical elements in the planning, any change could affect the overall planning and service quality would be poor or even risky. The uncertainty in travel times may be due to common factors such as varying road conditions, driving skills and weather conditions (Shi et al. 2019). However, the service time is not always fixed as estimated due to practical reasons, such as diagnosing time and parking situations (Shi et al. 2019).

Redjem and Marcon (2016) conducted a survey of many French home care structures and showed that patients need several care activities per day. Consequently, the synchronization of services will become essential, especially when more than one caregiver is assigned to provide one or multiple simultaneous service operations to the same patient. For example, an overweight patient will need two caregivers to lift him. It is not always easy to ensure the synchronization of service operations, this task will become very complicated if the number of caregivers is limited. Scheduling starting time of visits cannot be done independently from the other routes (Cissé et al. 2017), which increases the complexity of the problem.

Most previous efforts have been focused on studying the vehicle routing problem with stochastic parameters such as service time, demand and travel time (Laporte et al. 1992; Li et al. 2010; Taş et al. 2014a, b; Luo et al. 2016; Errico et al. 2016; Marinaki and Marinakis 2016; Mendoza et al. 2016; Chepuri and Homem-De-Mello 2005; Juan et al. 2013; Sarasola et al. 2016). But only a few studies have dealt with the HHCRSP with stochastic parameters (e.g., Shi et al. 2018, 2019; Yuan et al. 2015). These studies are limited to a single service operation per patient and, as, as far as we know, no researcher considered multiple services and their synchronization if they are required to be simultaneous for the stochastic HHCRSP. In this paper, we study the HHCRSP with stochastic travel and service times as well as multiple services per patient and their synchronization if they are required to be simultaneously performed. In addition, we propose a new stopping condition for the simulation, instead of just running it for a maximum number of iterations (100 iterations as in Shi et al. (2018)), to find a good estimation of the expected real value. It should be noted that in this paper, we extend our previous work (Bazirha et al. 2020), which is limited to a single service per patient, to cover multiple services and their synchronization. Also, we propose a heuristic based on a single solution (GVNS) to compare it with the performance of GA, which is a population based.

Several models (Charnes and Cooper 1959; Ben-Tal et al. 2009; Bernard 1955) have been proposed in the literature to deal with the uncertainty. The chance constrained model (Charnes and Cooper 1959) tries to find a solution for which the failure probability is less than some given threshold. In the failure case, the cost of corrective actions is not taken into consideration (Gendreau et al. 1996). In the robust optimization (Ben-Tal et al. 2009), the constructed solution is feasible for any realization of the uncertainty in a given set (Bertsimas et al. 2011). This model looks in a minimax fashion for the solution that provides the best "worst case" and solutions may be overly pessimistic. The uncertainty in this model is not stochastic, but rather deterministic and set-based (Bertsimas et al. 2011). The stochastic programming model with recourse (SPR model) (Bernard 1955) aims to find a first stage solution that minimizes the expected cost of the second stage solution, which is equal to the cost of the first stage solution plus the expected net cost of recourse (Gendreau et al. 1996). A recourse is introduced in the second stage if a solution goes against the constraints. The objective function of the SPR model is more meaningful than the chance constrained model (Gendreau et al. 1996). The SPR model is adopted to deal with stochastic parameters, the recourse is defined as a penalty cost for patients' delayed services and a remuneration for caregivers' extra working time.

We use CPLEX as an exact method, and the genetic algorithm (GA) as well as the general variable neighborhood search (GVNS) algorithms as approximate methods to solve the deterministic model. The performance of variable neighborhood search (VNS) (Mladenović and Hansen 1997) based heuristic is proved on the HHCRSP (Mankowska et al. 2014; Trautsamwieser and Hirsch 2011; Shi et al. 2019; Bazirha et al. 2019). Furthermore, the proposed solution coding takes into account both caregivers‘ routes and their assignment. VNS is more suitable since several neighborhood structures could be used to better explore the search space, which is not possible with heuristics designed to use a single neighborhood structure such as tabu search (TS) (Glover 1986) and simulated annealing (SA) (Kirkpatrick et al. 1983). We deal with stochastic travel and service times as well as the synchronization of multiple services, which makes caregivers' routes dependent. The simulation is used since computing the expected real value by an explicit mathematical formula is very complex. Classical solvers such as CPLEX and GUROBI are not suitable to be used with the simulation since their computation time increases monotonically with the problem size and the simulation takes time to converge towards the expected real value. The Genetic Algorithm (GA) based heuristic is adopted and combined with Monte Carlo simulation to solve the SPR model. Abundant experiments in the literature show that GA based heuristic has a good ability for global searching (Shi et al. 2018). In addition, parameters of GA are independent of problem parameters (e.g. number of services). In contrast, heuristics that extend and improve local search strategies such as TS, VNS and SA suffer from this problem, the number of neighborhoods increases with the problem size. At each time a neighborhood is browsed, the simulation is carried out to estimate the expected value of recourse, it requires time to find a good estimation. Indeed, as much as the number of iterations tends to infinity, the estimated value tends towards the expected real value (Law of large numbers). The GA is more suitable to solve the SPR model as its parameters are not depending on the problem size. The only parameter that could increase the complexity (number of running the simulation) is the population size. However, it is controllable and could be fixed independently of the problem size. In doing so, the complexity faced in heuristics extending local search methods will be avoided.

This paper is organized as follows: In Sect. 2, the state of the art of the HHCRSP is presented with a focus on papers dealing with stochastic models. Sect. 3 describes the problem statement. A two-stage stochastic programming model with recourse for the HHCRSP and Monte Carlo simulation are presented in Sect. 4. The heuristic methods used in the sequel, as well as the proposed procedure to ensure the synchronization of multiple services are presented in Sect. 5. Numerical experiments are conducted in Sect. 6. The paper is concluded in Sect. 7.

2 Literature review

HHC issues have attracted the attention of a large number of researchers in recent years. Their studies are not only from the perspective of social and medical techniques, but also from the optimization viewpoint (Shi et al. 2018). Mankowska et al. (2014) and Rasmussen et al. (2012) focused their works on temporal dependencies of services. In Mankowska et al. (2014), the authors proposed a mathematical model and a heuristic to deal with double synchronization as well as pairwise temporal precedence between jobs. In Rasmussen et al. (2012), the authors solved the problem by a dynamic column generation approach embedded into a branch-and-price framework, taking into consideration temporal dependencies between the start times of visits. Liu et al. (2017) dealt with the HHCRSP with lunch break requirements that has to be scheduled on caregivers' routes, they proposed a branch-and-price algorithm to solve the problem. Rest and Hirsch (2016) and Hiermann et al. (2015) considered the HHCRSP with multimodal transportation. Bazirha et al. (2019) studied the HHCRSP with multiple availability periods of patients. They proposed a mathematical model and a general variable neighborhood search based heuristic to minimize earliness and tardiness of services operations as well as caregivers' waiting times.

Deterministic models and methods are generally less robust. Without considering uncertainties, the quality of services would be very poor or risky. Laporte et al. (1992) worked on the vehicle routing problem with stochastic travel times. They proposed a stochastic programming model with recourse and a chance constrained model, which were solved by a general branch-and-cut algorithm. Li et al. (2010) studied the vehicle routing problem with stochastic travel and service times where each customer has a time window constraint. They proposed a stochastic programming model with recourse and a chance constrained model, which were solved by a tabu search based heuristic. Taş et al. (2014a) focused their work on time-dependent vehicle routing problem with soft time windows and stochastic travel times. The authors suggested heuristics based on tabu search and adaptive large neighborhood search to minimize the total weighted cost, which includes service costs incurred due to earliness and tardiness of services, vehicles used, transportation, and overtime. They also used another solution approach to solve the problem using branch-and-price and column generation algorithms Taş et al. (2014b). Yuan et al. (2015) worked on home health care scheduling and routing problem with skill requirements and stochastic service times. They implemented a branch-and-price algorithm to optimize transportation cost, penalty for tardiness of customers' services, caregivers' fixed cost and service cost. Luo et al. (2016) implemented an adaptive large neighborhood search based heuristic to solve the vehicle routing problem with stochastic demands and weight-related costs. The authors combined dynamic programming recursions with the proactive recourse strategy to compute the expected cost for each route. Errico et al. (2016) used a priori optimization with recourse for the vehicle routing problem with hard time windows and stochastic service times. The authors proposed two recourse policies for infeasible routes: skipping the service at the current customer and skipping the visit at the next customer. They solved the problem using brunch-cut-and-price algorithms. Marinaki and Marinakis (2016) studied the vehicle routing problem with stochastic demands and proposed a glowworm swarm optimization algorithm as a solving approach. Mendoza et al. (2016) worked on the vehicle routing problem with stochastic demands and duration constraints. They proposed a chance constrained model, which was solved by a greedy randomized adaptive search procedure enhanced by heuristic concentration. Shi et al. (2018) studied the simultaneous delivery and pick-up problem with stochastic travel and service times in HHC. The authors adopted a stochastic programming model with recourse to minimize the transportation cost and the expected value of recourse caused by patients' delayed services and caregivers' extra working time. They solved the deterministic model using Gurobi solver and heuristics based on simulated annealing, hybrid genetic algorithm, firefly algorithm and bat algorithm. The SPR model was solved by a simulated annealing based heuristic. Shi et al. (2019) used the robust optimization to deal with the HHCRSP with caregivers' qualification, a maximum of patients to visit per caregiver and stochastic travel and service times. They implemented heuristics based on simulated annealing, variable neighborhood search and tabu search to solve the problem.

As presented above, on the one hand, exact approaches such as branch-and-price algorithm have been proposed to solve these problems, where the service time (Errico et al. 2016; Yuan et al. 2015) or the travel time (Taş et al. 2014b) is considered stochastic and the expected value is computed exactly by a mathematical formula. On the other hand, in Shi et al. (2018) the problem has been considered with travel and service times that are supposed stochastic, and the simulation has been used to estimate the expected value. To solve our problem within a reasonable computational time, a heuristic is used rather than exact approaches. Most of previous efforts have been focused on studying the vehicle routing problem, but there are only a few studies that have dealt with the HHCRSP with stochastic parameters (e.g., Shi et al. 2018, 2019; Yuan et al. 2015). However, these studies have been generally deterministic if the patients' requested services are considered multiple such as (Rasmussen et al. 2012; Mankowska et al. 2014; Redjem and Marcon 2016).

In this work, we combine three challenges that make the problem very complex to solve:

- The HHCRSP is NP-hard problem and difficult to solve for large instances since it is a combination of two NP-hard problems: the vehicle routing problem with time windows (Bräysy and Gendreau 2005) and the nurse rostering problem (Burke et al. 2004);
- The uncertainty of service and travel times requires evaluating the expected value;
- The synchronization of multiple services that increases the difficulty of the problem due to the routes that could not be evaluated independently.

3 Problem statement

The set of patients is denoted by $N = \{1, 2, 3, ..., n\}$, where n is the number of patients dispersed in a geographic area. For each patient i, a preferred time window is defined as $[a_i, b_i]$, where a_i and b_i are, respectively, the earliest and latest possible service times of the visits. The patient can request one or more service operations, which should be synchronized if they are required to be simultaneous. Each service operation s has a duration $\widetilde{t_{is}}$, which is stochastic and without preemption. The travel time from patient i to patient j is supposed stochastic and denoted $\widetilde{T}_{ij}$.

Since caregivers do not all have the same skills, the HHC company must assign them to patients according to the skills required by the nature of the requested services. The caregivers set is denoted by $K = \{1, 2, 3, ..., c\}$, where c is their number and the set of services is denoted by $S = \{1, 2, 3, ..., q\}$, where q is the maximum number of services that the HHC company could provide. The duty length for each caregiver k is given by $[d_k, e_k]$, where d_k and e_k are, respectively, the earliest and latest service times

Patients' requested services are indicated in a matrix of binary parameters δ_{is}. If a patient $i \in N$ requests a service operation $s \in S$, the binary parameter δ_{is} is set to 1 and 0 otherwise. Accordingly, caregivers' qualifications are indicated in a matrix of binary parameters Δ_{ks}. If the caregiver $k \in K$ is skilled to provide the service operation $s \in S$, the binary parameter Δ_{ks} is set to 1 and 0 otherwise. Each service operation must be performed by a single caregiver for patients requesting multiple services.

The problem is to define a daily planning in order to minimize the transportation cost and the expected value of recourse caused by patients' delayed services and caregivers' extra working time. Each patient receives qualified caregivers to perform his requested services within his preferred availability period. If there are multiple services that are required to be simultaneously performed, they must be synchronized. Caregivers leaving from the HHC center, to serve assigned patients, return to the center.

4 Mathematical formulation

The problem is formulated as a two-stage stochastic programming model with recourse. The first stage aims to compute the transportation cost while the second stage is to introduce the recourse, which is defined as a penalty cost for patients' delayed services and a remuneration for caregivers' extra working time. The notation of sets, decision variables and parameters used in the model are defined as follows:

4.1 Sets

- N: set of patients;
- N^0 and N^{n+1}: set of patients including the HHC center, which is represented by artificial nodes 0 and $n+1$ where $N^0 = N \cup \{0\}$ and $N^{n+1} = N \cup \{n+1\}$;
- K: set of caregivers;
- S: set of services and skills.

4.2 Deterministic parameters

- n: number of patients;
- c: number of caregivers;
- q: number of services (skills);
- M: large positive number;
- $[a_i, b_i]$: patients' time windows;
- $[d_k, e_k]$: caregivers' time windows;
- c_{ij} : transportation cost between patients i and j;

- δ_{is}: equals 1 if the patient $i \in N$ requires the service operation $s \in S$;
- $\varDelta_{ks}$: equals 1 if the caregiver $k \in K$ is qualified to provide the service operation $s \in S$;
- λ_i: equals 1 if service operations requested by the patient i must be simultaneous.

4.3 Stochastic parameters

- $\widetilde{T}_{ij}$: travel time from the patient i to the patient j;
- $\widetilde{t}_{is}$: processing time of the service operation s at the patient $i \in N$;
- $\mathbb{E}(.)$: the expected value of the second stage of the model, which expresses the recourse value caused by extra working times and delayed services operations.

4.4 Decision variables

- x_{ijk}: equals 1 if the caregiver k visits the patient j after the patient i, 0 otherwise;
- y_{iks}: equals 1 if the service operation s is provided by the caregiver k to the patient i, 0 otherwise;
- $\widetilde{S}_{ik}$: start time of a service operation at the patient i provided by the caregiver k.

4.5 Parameters for recourse model

- v_i: tardiness of a service operation at the patient i ;
- o_k: extra working time for the caregiver k;
- α: unit penalty cost for a tardiness of a service operation;
- γ: caregiver's remuneration unit for an extra working time.

4.6 Mathematical model

The stochastic programming model with recourse formulation proposed to solve the HHCRSP with stochastic travel and service times, which is adapted from our previous work (Bazirha et al. 2019) and the recourse model from Shi et al. (2018) by adding other constraints of the HHC context (multiple services and synchronization), is defined as follows:

$$\min Z = \sum_{k=1}^{c}\sum_{i=0}^{n}\sum_{j=1}^{n+1} c_{ij}x_{ijk} + \mathbb{E}\left[\min\left(\alpha\sum_{i=1}^{n} v_i + \gamma\sum_{k=1}^{c} o_k\right)\right]$$

s.t.

$$\sum_{i=0}^{n}\sum_{k=1}^{c} x_{ijk} = \sum_{s=1}^{q} \delta_{js} \qquad j \in N \tag{1}$$

$$\sum_{j=1}^{n+1}\sum_{k=1}^{c} x_{ijk} = \sum_{s=1}^{q} \delta_{is} \qquad i \in N \tag{2}$$

$$\sum_{i=0}^{n} x_{i,n+1,k} = 1 \qquad k \in K \tag{3}$$

$$\sum_{j=1}^{n+1} x_{0,j,k} = 1 \qquad k \in K \tag{4}$$

$$\sum_{i=0}^{n} x_{imk} = \sum_{j=1}^{n+1} x_{mjk} \qquad m \in N, k \in K \tag{5}$$

$$\widetilde{S}_{ik} + \sum_{s=1}^{q} \widetilde{t}_{is} y_{iks} + \widetilde{T}_{ij} \leq \widetilde{S}_{jk} + (1 - x_{ijk})M \qquad i \in N^0, j \in N^{n+1}, k \in K \tag{6}$$

$$\widetilde{S}_{ik} \leq \sum_{s=1}^{q} y_{iks} M \qquad i \in N, k \in K \tag{7}$$

$$\sum_{j=1}^{n+1} x_{ijk} = \sum_{s=1}^{q} y_{iks} \qquad i \in N, k \in K \tag{8}$$

$$2y_{iks} \leq \delta_{is} + \Delta_{ks} \qquad i \in N, s \in S, k \in K \tag{9}$$

$$\sum_{s=1}^{q} y_{iks} \leq 1 \qquad i \in N, k \in K \tag{10}$$

$$\sum_{k=1}^{c} y_{iks} \leq 1 \qquad i \in N, s \in S \tag{11}$$

$$d_k \leq \widetilde{S}_{0k} \qquad k \in K \tag{12}$$

$$\widetilde{S}_{(n+1)k} \leq e_k + o_k \qquad k \in K \tag{13}$$

$$\left(\sum_{s=1}^{q} y_{iks} - 1\right) M + a_i \leq \widetilde{S}_{ik} \qquad i \in N, k \in K \tag{14}$$

$$\widetilde{S}_{ik} + \sum_{s=1}^{q} \widetilde{t}_{is} y_{iks} \leq b_i + v_i + (1 - \sum_{s=1}^{q} y_{iks})M \qquad i \in N, k \in K \tag{15}$$

$$\sum_{v=1}^{c} \widetilde{S}_{iv} - \sum_{s=1}^{q} \delta_{is} \widetilde{S}_{ik} \leq (2 - \lambda_i - \sum_{s=1}^{q} y_{iks})M \qquad i \in N, k \in K \tag{16}$$

$$\sum_{v=1}^{c} \widetilde{S}_{iv} - \sum_{s=1}^{q} \delta_{is} \widetilde{S}_{ik} \geq (\sum_{s=1}^{q} y_{iks} + \lambda_i - 2)M \qquad i \in N, k \in K \tag{17}$$

$$x_{iik} = 0 \qquad i \in N, k \in K \tag{18}$$

$$\widetilde{S}_{ik} \geq 0 \qquad i \in N, k \in K \tag{19}$$

$$v_i \geq 0 \qquad i \in N \tag{20}$$

$$o_k \geq 0 \qquad k \in K \tag{21}$$

$$x_{ijk} \in \{0, 1\} \qquad i \in N, j \in N, k \in K \tag{22}$$

$$y_{iks} \in \{0, 1\} \qquad i \in N, k \in K, s \in S \tag{23}$$

The objective function is to minimize the total transportation cost and the expected value of recourse caused by patients' tardiness of services operations and caregivers' extra working times. Constraints (1) and (2) state that each patient will be visited by a group of caregivers, the number will depend on the kind of requested services. Constraints (3) and (4) state that each caregiver leaving the HHC center to visit assigned patients must get back there. Constraints (5) impose route continuity for the patients assigned to a caregiver k. In doing so, tours will be constructed rather than open paths. Constraints (6) determine the service operations' starting time of the patient j with respect to service operations' completion time of the patient i.

These constraints require that the starting time of services along the route of a caregiver are strictly increasing. Under this condition, these constraints can also remove sub-tours because a return to an already visited patient will violate the start time of the previous visit (Mankowska et al. 2014). Constraints (7) initialize the starting time to zero if the caregiver k is not assigned to the patient i. Constraints (8) define the variables y_{iks}, the caregiver k is assigned to the patient i if after having visited the patient i he will visit the next patient j or return to the HHC center. Constraints (9) ensure the qualification of the caregiver k to provide the service operation s requested by the patient i. Constraints (10) ensure that each caregiver provides a single service operation to an assigned patient i. Constraints (11) guarantee that for a patient i, each service operation is provided by only one caregiver. Constraints (12) and (13) ensure the respecting of caregivers' time windows. Constraints (14) and (15) guarantee the respecting of patients' availability periods. Constraints (16) and (17) ensure caregivers' starting time synchronization if a patient's requested services must be simultaneous ($\lambda_i = 1$). Constraints (18)–(23) are the domains of the decision variables.

4.7 The expected recourse estimation procedure

The recourse model in stochastic programming is formulated by different ways depending on the nature of the problem. The recourse will be introduced if a solution goes against the constraints. On the VRP with capacity (CVRP) and stochastic demand, the recourse could be defined as returning to the depot if a vehicle is filled and assigned customers are not all visited. This recourse could be applied only to CVRP. In Errico et al. (2016), recourse policies are defined as: skipping the service at the current customer and skipping the visit at the next customer. The main drawback of this recourse is to decrease patients' satisfaction since their requested services could be ignored. In our problem, the recourse model considered in Shi et al. (2018) is adopted, since we consider that all patients' requested services will be provided. Constraints related to travel and service times will be relaxed to soft ones, and recourse will be introduced if a solution goes against these constraints. The recourse is defined as: a penalty cost for a tardiness of a service operation and a remuneration for an extra working time.

Monte Carlo simulation (Von Neumann and Ulam 1951) is a method for estimating a numerical quantity that uses random numbers. Computing the real expected value ($\mathbb{E}(.)$) is very complex since caregivers' routes are dependent and travel and service times are considered stochastic. This simulation is used as an alternative way to compute the estimated value of recourse ($\widehat{\mathbb{E}(.)}$). $\widehat{\mathbb{E}(.)}$ gives the average recourse that will be incurred for a given schedule. A robust planning will be performed since different scenarios that might occur have been simulated. The recourse of tardiness of services operations and caregivers' overtimes working that might arise will be approximated by $\widehat{\mathbb{E}(.)}$.

Algorithm 1 is performed to estimate the expected value of recourse ($\widehat{\mathbb{E}(.)}$). sum_T is the total tardiness of services operations occurred at patients. sum_O is the total extra working times of caregivers and SS_i is the synchronized starting time for the patient i. Given a solution, patients are assigned to caregivers and caregivers' routes (order of visiting) are defined. Firstly, caregivers' transportation cost are computed, which are independent of travel and service times. Secondly, Monte Carlo simulation 1 is used to estimate the expected value of recourse caused by patients' delayed services and caregivers' extra working time. At each iteration, caregivers' travel times $\widetilde{T}_{ij}$ and patients' service times $\widetilde{t}_{is}$ are randomly generated, then the synchronization of services operations will be ensured. The estimation procedure is running until either condition 1 or condition 2 is met. Condition 1 is fixed as a maximum number of iterations, denoted by $MaxIterMCS$, and condition 2 is defined as follows:

Algorithm 1: The expected recourse estimation procedure

```
Initialization: ;
    - Set $iter = 0$ ;
    - $V_k$: set of patients assigned to the caregiver k ;
while (condition 1 or condition 2 ) do
    Generate randomly $\widetilde{T}_{ij}$ ;
    Generate randomly $\widetilde{t}_{js}$ ;
    while ($SS_j$ is changing ) do
        Set $sum_T = 0$ and $sum_O = 0$;
        for $k \leftarrow 1$ to $K$ do
            for $j \in V_k$ do
                Set $v_j = 0$ ;
                if $\lambda_j = 0$ then
                    Calculate $\widetilde{S}_{jk}$ ;
                    if $\widetilde{S}_{jk} + \widetilde{t}_{js} > b_j$ then
                        Set $v_j \longleftarrow \widetilde{S}_{jk} + \widetilde{t}_{js} - b_j$ ;
                    end
                else
                    Calculate $\widetilde{SS}_j$ ;
                    if $\widetilde{SS}_j + \widetilde{t}_{js} > b_j$ then
                        Set $v_j \longleftarrow \widetilde{SS}_j + \widetilde{t}_{js} - b_j$ ;
                    end
                end
                $sum_T \longleftarrow sum_T + v_j$ ;
            end
            Set $o_k = 0$ ;
            if $\widetilde{S}_{(N+1)k} > e_k$ then
                Set $o_k \longleftarrow \widetilde{S}_{(N+1)k} - e_k$ ;
            end
            $sum_O \longleftarrow sum_O + o_k$ ;
        end
    end
    $iter \longleftarrow iter + 1$ ;
end
Set $\widehat{\mathbb{E}(.)} \longleftarrow \frac{\alpha \times sum_T + \gamma \times sum_O}{iter}$ ;
```

$gap = \left| \frac{E(.)_{(Iter-1)} - E(.)_{Iter}}{E(.)_{(Iter-1)}} \right| < \epsilon$. This condition means that the gap must exactly hold at a maximum number of iterations, denoted by $MaxIterGap$, and it expresses the gap between $E(.)_{(Iter-1)}$ and $E(.)_{Iter}$ with an error ϵ. In other words, the estimated value is considered close to the real value when the formula is satisfied ($gap < \epsilon$) for the fixed maximum number of iterations, which means that $E(.)_{Iter}$ are close to each other for the last $MaxIterGap$ iterations. The counter is set to zero if the formula is not satisfied for an iteration. The more ϵ is reduced to 0 and $MaxIterGap$ increases, the more the estimated value tends towards the expected real value. However, the computational time increases significantly given that for each individual the expected value is calculated. Condition 1, i.e. $MaxIterMCS$, is used to avoid falling into an infinite loop that could be caused by the condition 2, especially when ϵ tends towards 0 and $MaxIterGap$ tends towards a large number. The condition used in Shi et al. (2018), is a particular case of the proposed conditions ($\epsilon = 0$, $MaxIterGap = +\infty$ and $maxIterMCS = 100$).

Table 1 Example of solution encoding

Patients	5 (2)	2 (1)	6 (2)	3 (1)	1 (2)	4 (3)	1 (3)
Caregivers	2	1	1	2	1	2	2

5 Approximate methods

5.1 Encoding

To take into consideration both assignment and routing of caregivers, a solution will be represented by two chromosomes where their sizes equal to the number of the total requested services of all patients. The first one (patients' chromosome) will contain patients and requested services (are included in parenthesis), and describes patients' visiting order. The second (caregivers' chromosome) will contain assigned caregivers. To deal with multiple services, we will duplicate a patient i as many times as the number of services he requested.

Example Consider 6 patients and 2 caregivers skilled to provide 3 types of services operations. The patient 1 requests two service operations 2 and 3. A solution will be encoded as follows (see Table 1).

The caregiver 1 will visit patients 2, 6 and 1 to provide respectively service operations 1, 2 and 2.

5.2 Decoding

Given a solution encoded as proposed above. For each subset of patients assigned to a caregiver, arrival and starting times will be iteratively calculated in the same order as they appear in patients' chromosome. Due to dependent caregivers' routes, the starting times must be synchronized for patients who requested multiple simultaneous services ($\lambda_i = 1$). Otherwise ($\lambda_i = 0$), each caregiver arrives will start providing the assigned service operation and leave when finishing. In the case $\lambda_i = 1$, a synchronized starting time SS_i is considered and it is equal to the maximum between the earliest time window a_i for the patient i and caregivers' arrival times $\{A_{ik_1}, A_{ik_2}, ...\}$. Algorithmically, arrival and starting times will be iteratively calculated in the same order as they appear in the matrix. Each time a caregiver k arrives to a patient i, SS_i is first initialized by a_i and then will take the value A_{ik} if ($A_{ik} > SS_i$). These steps are repeated until SS_i will not change for each patient. In the end, caregivers' starting times at patients will be the synchronized starting time $S_{ik} = SS_i$.

Remark The decoding method could fall into an infinite loop, so service operations would never be synchronized and the solution will be consequently considered infeasible as illustrated by the example of Table 2. Suppose that patients 1 and 4 request two services 3 and 2. Hence, services requested for both patients 1 and 4 cannot be synchronized because caregiver 1 visits patient 1 then patient 4, and caregiver 2 visits patient 4 then patient 1. In the case of hard/fixed time windows, the infeasibility will be considered if after a number of iterations, the synchronized starting time value for a patient i (in the case $\lambda_i = 1$) will be greater than the latest service time b_i. However, in the case of soft/flexible time windows, the solution infeasibility is considered when the number of iterations exceeds the maximum number of iterations denoted by $MaxIterSyn$.

Table 2 Example of infeasible solution under the synchronization constraint

Patients	5 (2)	2 (1)	6 (2)	3 (1)	**1 (2)**	**4 (3)**	**4 (2)**	**1 (3)**
Caregivers	2	1	1	2	**1**	**2**	**1**	**2**

5.3 Genetic algorithm

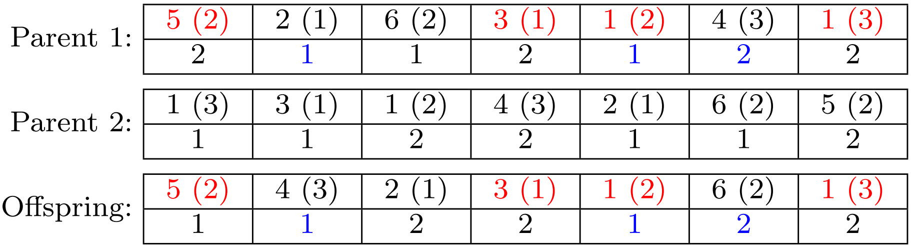
The main drawback of exact methods is the allowed running time, which is not always enough even to find a feasible solution, especially for NP-hard problems. Several heuristics have been proposed and are able to yield near-optimal solutions to hard problems in a reasonable amount of time such as GA, which was invented by John Holland in the 1960s (Holland 1992). Abundant experiments in the literature show that GA based heuristics have a good ability for global searching (Shi et al. 2018), which involve a simulation of Darwinian "survival of the fittest". This simulation takes an initial population and through the mechanism of reproduction (selection, crossover and mutation), the produced offspring inherits parents' characteristics and will be added to the next generation. This process keeps on iterating and the fittest individuals will be kept in the end.

5.3.1 Crossover operator

The crossover operation is the main genetic operator in GA used to combine the genetic information of two parents to stochastically generate new offspring. A large number of crossover operators have been proposed such as 1-point crossover, 2-point crossover and uniform order crossover (UOX). These 3 crossover operators were implemented and tested on some instances to choose the best one, the UOX operator, was developed by David Davis (1991), gives better results. The UOX preserves the position of some genes and the relative ordering of the rest. After the selection of two parents is done, the two chromosomes are separated and the UOX operator is independently applied for each one (patients' chromosome and caregivers' chromosome) and then are recombined. A function is used to repair infeasible offspring according to assignment constraints. The UOX operator is applied as follows:

1. For each parent chromosome a binary string of the same length is randomly generated;
2. The intermediate offspring preserves genes of the first parent where the generated string contains **"1"**;
3. Sort genes not preserved in the first parent in the same order as they appear in the second parent. For patients' chromosome, two genes are considered similar if patient's number and the requested service of each gene are equal. Replace genes not preserved in the first parent by genes of second parent for caregivers' chromosome.

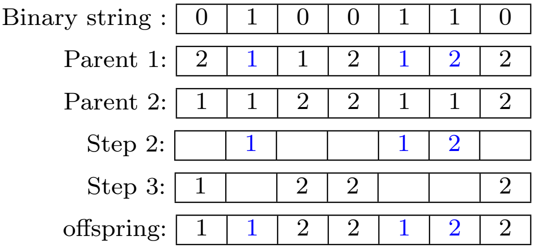
Steps 1 and 2 are similar for both chromosomes (caregivers and patients). However, step 3 is adapted for each chromosome. Since the repetition of caregivers does not pose a problem, genes not preserved in first parent are replaced by genes of the second parent (see Figs. 1 and 2). In contrast, each patient with the requested service operation must appear only once in the solution, sorting genes not preserved in the first parent in the same order as they appear in the second parent is mandatory to avoid the duplication or the deletion of a patient from the solution (see Figs. 1 and 3).



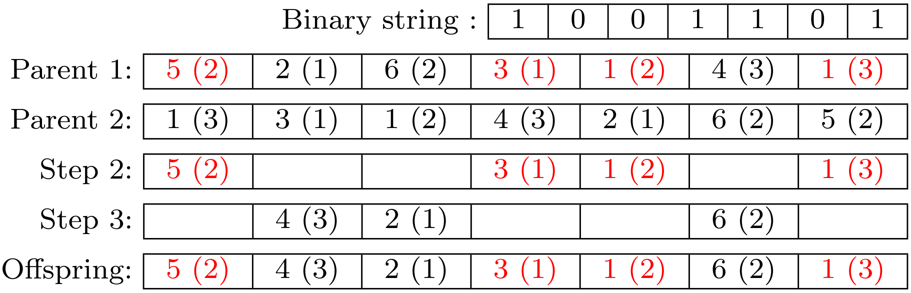
Parent 1:	5 (2)	2 (1)	6 (2)	3 (1)	1 (2)	4 (3)	1 (3)
	2	1	1	2	1	2	2
Parent 2:	1 (3)	3 (1)	1 (2)	4 (3)	2 (1)	6 (2)	5 (2)
	1	1	2	2	1	1	2
Offspring:	5 (2)	4 (3)	2 (1)	3 (1)	1 (2)	6 (2)	1 (3)
	1	1	2	2	1	2	2

Fig. 1 Example of parents and an offspring

Fig. 2 Example of caregivers' crossover operator



Binary string :	0	1	0	0	1	1	0
Parent 1:	2	1	1	2	1	2	2
Parent 2:	1	1	2	2	1	1	2
Step 2:		1			1	2	
Step 3:	1		2	2			2
offspring:	1	1	2	2	1	2	2



Binary string :	1	0	0	1	1	0	1
Parent 1:	5 (2)	2 (1)	6 (2)	3 (1)	1 (2)	4 (3)	1 (3)
Parent 2:	1 (3)	3 (1)	1 (2)	4 (3)	2 (1)	6 (2)	5 (2)
Step 2:	5 (2)			3 (1)	1 (2)		1 (3)
Step 3:		4 (3)	2 (1)			6 (2)	
Offspring:	5 (2)	4 (3)	2 (1)	3 (1)	1 (2)	6 (2)	1 (3)

Fig. 3 Example of patients' crossover operator

Table 3 Example of mutation operators

Patients	5 (2)	2 (1)	**1 (2)**	3 (1)	**6 (2)**	4 (3)	1 (3)
Caregivers	**1**	1	1	2	1	2	2

5.3.2 Mutation operator

The role of the mutation operator, which is randomly performed with a small probability, is to avoid being trapped in local optima, to excavate the diversity of the individuals in the population and to diversify the search directions. Two mutation operators are proposed, patients' mutation operator is to exchange two patients' positions randomly generated, and caregivers' mutation operator is to switch an assigned caregiver to a patient randomly. Taking the example of the Table 1 and supposing that the probabilities of carried out mutation operators are satisfied for both chromosomes. The position for caregivers' chromosome is 1 and positions for patients' chromosome are 3 and 5. Table 3 shows the obtained solution after carry out the mutation operators.

5.3.3 Fitness and Selection

In selection phase, the fittest individuals are selected for reproduction, which let them pass their genes to the next generation. The most commonly known selection methods include roulette rank selection, wheel selection, and tournament selection (Goldberg and Deb 1991). The first one suffers from the slow convergence and the sorting is done for the population to assign ranks, which increase the computational time. The second suffers from problem of premature convergence due to the possible presence of dominant individuals that always win the competition and are selected as a parent, which will be rewarded with a large number of offspring in the next generation while the population size is kept constant (Baker 1985).

In this study, the tournament selection will be used. It consists of selecting some individuals from the population, then those individuals compete against each other. The one with the highest fitness wins and participates in reproduction. Two fitness functions are used to evaluate solutions, F_D is used for the deterministic model and F_{SPR} is used for the SPR model and are defined as follows:

$$F_D = \sum_{k=1}^{c} \sum_{i=0}^{n} \sum_{j=1}^{n+1} c_{ij} x_{ijk} + \beta \left(T_s + O_c \right) \tag{24}$$

$$F_{SPR} = \sum_{k=1}^{c} \sum_{i=0}^{n} \sum_{j=1}^{n+1} c_{ij} x_{ijk} + \mathbb{E}(.) \tag{25}$$

Infeasible solutions according to time windows constraints are accepted with a penalty cost, which is the sum of tardiness of services operations T_s and caregivers' overtimes O_c. The coefficient β is introduced to ensure the convergence of infeasible solutions to feasible ones for the deterministic model, the more β is higher the more penalized solutions are to be eliminated from the population in selection step. The individual with small fitness will be selected.

5.3.4 Genetic algorithm procedure

At each time crossover and mutation operations are applied to the selected parents, the generated offspring might not be feasible. A service operation could be assigned to a caregiver who is not qualified to provide. To avoid this issue, the repair function is called after applying crossover and mutation operations. This function checks each assigned service to a caregiver and verify if he is qualified to provide it, otherwise, a randomly qualified caregiver is selected.

5.3.5 Initial population

The initial population for a given size P_{size} is randomly generated as follows:

- For $i = 1$ to P_{size} do:
- Generate a random visiting order (see Table 4);
- For each patient, assign a qualified caregiver randomly selected (see Table 5);
- Calculate the transportation cost and the expected value of recourse for the stochastic model by Eq. (25). For the deterministic model, the transportation cost, penalized tardiness of services operations and caregivers' extra working times are computed by Eq. (24).

Algorithm 2: GA procedure

```
Initialization: ;
    - Define $P_{size}$: population size;
    - Define $P_c$: crossover probability ;
    - Define $P_S$: mutation probability ;
    - Define $P$: population ;
    - Define $O_{ff}$: Offspring ;
while (the stopping condition is not reached) do
    for $i \leftarrow 1$ to $P_{size}$ do
        Selection: $P_1$ and $P_2$ ;
        Generate randomly $p_1$ ;
        if $p_c < p_1$ then
            Set $Off \longleftarrow Crossover(P_1, P_2)$ ;
        end
        Generate randomly $p_2$ ;
        if $p_m < p_2$ then
            Set $Off \longleftarrow Mutation(Off)$ ;
        end
        Repair $Off$ ;
    end
    Set $P \longleftarrow Off$ ;
end
```

Table 4 Example of visiting order

Patients	5 (2)	1 (3)	3 (1)	1 (2)	4 (3)	2 (1)	6 (2)
Caregivers							

Table 5 Example of caregivers' assignment to patients

Patients	5 (2)	1 (3)	3 (1)	1 (2)	4 (3)	2 (1)	6 (2)
Caregivers	1	2	2	1	1	2	1

5.4 Variable Neighborhood Search

The drawback of local search strategies is known as falling into a local optimum with a poor value. Several heuristics, extend and improve the local search strategies, have been proposed to avoid being trapped in a local optima such as tabu search (Glover 1986), simulated annealing (Kirkpatrick et al. 1983) and variable neighborhood search (VNS) (Mladenović and Hansen 1997). VNS is based on the idea of systematic changes of neighborhoods in the search to find a better solution. When an initial solution is defined, the VNS proceeds by a descent method exploring the predefined neighborhoods of a solution to find a local minimum. Each time the descent method is trapped in a local optimum, the shaking phase is randomly applied to generate a new solution and start over the search. Many versions of VNS are used in the literature such as: (1) variable neighborhood descent (VND) is a deterministic version of VNS, the defined neighborhoods are applied to the initial solution in a predefined order, the searching restart from the first neighborhood when a new local minimum is found. (2) reduced variable neighborhood search is a pure stochastic search method where only the shaking phase is applied to solutions; (3) general variable neighborhood search (GVNS) is a VNS where the local descend method is replaced by the VND.

Original solution :

Patients	5 (2)	2 (1)	6 (2)	3 (1)	1 (2)	4 (3)	1 (3)
Caregivers	2	1	1	2	1	2	2

Neighbor solution :

Patients	5 (2)	2 (1)	6 (2)	3 (1)	1 (2)	4 (3)	1 (3)
Caregivers	1	1	1	2	1	2	2

Fig. 4 Example of switch neighborhood moves

Original solution :

Patients	5 (2)	2 (1)	6 (2)	3 (1)	1 (2)	4 (3)	1 (3)
Caregivers	2	1	1	2	1	2	2

Neighbor solution :

Patients	5 (2)	2 (1)	6 (2)	3 (1)	1 (2)	4 (3)	1 (3)
Caregivers	2	1	2	2	1	1	2

Fig. 5 Example of inter-swap neighborhood moves

5.4.1 Neighborhoods

A neighborhood relation is defined on the search space to generate neighborhoods of a solution, and it is a transformation applied to a solution in order to generate new candidate solutions. Four neighborhood relations are defined, two neighborhoods are used to intensify caregivers' assignment and the two others are used to intensify the routes (order of visiting), intra-shift, inter-swap and intra-swap neighborhoods are adopted from Mankowska et al. (2014) and adapted to the proposed decoding. The total requested services of all patients is denoted by s.

1. Switch: this neighborhood is defined as a reassignment of a patient i to another caregiver k, it takes a patient i out of a given caregiver's route and inserts him into another caregiver's route (see Fig. 4). The size of possible neighborhoods is less than $s(c-1)$ depending on caregivers' skills. The equality will be held if all caregivers are skilled to provide all services operations;
2. Inter-swap: the neighborhood aims to change caregivers' assignment to patients. Given two patients, assigned caregivers are swapped. This move interchanges two patients that are assigned to two different routes (see Fig. 5). The size of possible neighborhoods equals to $\frac{(s-1)\times s}{2}$;
3. Intra-shift: given a solution and a patient i, the neighborhood is defined as shifting this patient with the assigned caregiver to another position. This move will change assigned patients' order for the caregiver k (see Fig. 6). The size of possible neighborhoods equals to $(s-1)\times s$;
4. Intra-swap: given a solution and two patients i and j, the neighborhood is defined as exchanging of their positions with preserving the assigned caregivers. This move will change the order of patients' visiting for the both assigned caregivers (see Fig. 7). The size of possible neighborhoods equals to $\frac{(s-1)\times s}{2}$.

Original solution :

Patients	**5 (2)**	2 (1)	6 (2)	3 (1)	1 (2)	4 (3)	1 (3)
Caregivers	**2**	1	1	2	1	2	2

Neighbor solution :

Patients	2 (1)	6 (2)	3 (1)	1 (2)	**5 (2)**	4 (3)	1 (3)
Caregivers	1	1	2	1	**2**	2	2

Fig. 6 Example of intra-shift neighborhood moves

Original solution :

Patients	5 (2)	2 (1)	**6 (2)**	3 (1)	1 (2)	**4 (3)**	1 (3)
Caregivers	2	1	**1**	2	1	**2**	2

Neighbor solution :

Patients	5 (2)	2 (1)	**4 (3)**	3 (1)	1 (2)	**6 (2)**	1 (3)
Caregivers	2	1	**2**	2	1	**1**	2

Fig. 7 Example of intra-swap neighborhood moves

5.4.2 Shaking

The shaking phase is the core of the algorithm, which is called whenever the local descent methods are trapped in a local optimum (Mladenović and Hansen 1997). It is defined as a series of moves applied to a solution to skip from a local optima and jump to another solution. The four proposed neighborhoods are used as operators for the shaking phase ($k_{max} = 4$), which will be applied a number of times, each time the move is randomly generated.

5.4.3 Local search methods

Local descend methods browse candidate solutions to improve the incumbent solution. Two algorithms could be used to select the next solution (Taillard 1990): 1) the first improvement consists of selecting the first neighbor that improves the incumbent solution; 2) the best improvement consists selecting the best candidate solution after that all neighbors are tested to choose the best. Four local descend methods will be used, each one is matched to a neighborhood of the four proposed. The best improvement algorithm will be adopted.

5.4.4 Initial solution

The initial solution is randomly generated the same way for the GA based heuristic by setting $P_{size} = 1$.

5.4.5 GVNS procedure

The VND is applied to each solution x' generated in the shaking phase. Candidate neighborhoods generated by switch and inter-swap operators could be infeasible, a caregiver could be assigned to provide a service operation that he is not qualified to provide. To avoid this issue, for switch operator only feasible assignments are accepted. For inter-swap operator, infeasible solutions caused by patients' interchange are ignored. If the solution x (resp. x') is improved, the k (resp. l) is initialed to 1 (see Algorithm 3).

Algorithm 3: GVNS procedure

```
Initialization: ;
    - Set $K_{max} = 4$ and $l_{max} = 4$ ;
    - Generate an initial solution $x$ ;
while (the stopping condition is not reached) do
    for $k \leftarrow 1$ to $k_{max}$ do
        Generate at random $x'$ in the $k^{th}$ Neighborhood of $x$, the $k^{th}$ Neighborhood move is applied $h$
          times to $x$ ;
        for $l \leftarrow 1$ to $l_{max}$ do
            Find the best neighbor $x''$ of $x'$ in $N_l(x')$ ;
            if $f(x'') < f(x')$ then
                Set $x' \longleftarrow x''$ and $l \longleftarrow 1$ ;
            else
                Set $l \longleftarrow l + 1$ ;
            end
        end
        if $f(x') < f(x)$ then
            Set $x \longleftarrow x'$ and $k \longleftarrow 1$ ;
        else
            Set $k \longleftarrow k + 1$ ;
        end
    end
end
```

5.5 Tuning parameters

A series of tests are performed to find the best values of tuning parameters (see Table 6). The stopping criterion for both GA and GVNS based heuristics is set as the number of no improvement in the best solution found for a maximum number of iterations. The total requested services of all patients is denoted by s.

6 Numerical experiments

The experiments run under windows 10 on the computer with Intel i7-7600U 2.80-GHz CPU and 16 GB of RAM. The reduced SPR model to the Mixed Integer Linear Program (MILP) is implemented and tested using CPLEX version 12.8. GA and GVNS based heuristics are coded and tested using the language C++.

6.1 Test instances

The test instances have been randomly generated using the benchmark instances from Mankowska et al. (2014). The HHC center and patients' locations are placed at random positions in the area of 100 × 100 distance units. Travel time T_{ij} and cost c_{ij} are equal to the Euclidean distance d_{ij} between patients' locations truncated to an integer. In practice, cost and travel time between patients are quasi-proportional to the distance. It is assumed that $T_{ij} = coef_1 \times d_{ij}, c_{ij} = coef_2 \times d_{ij}$ and $coef_1 = coef_2 = 1$. Service times t_{is} are randomly generated within an interval of [10, 20] minutes. Six types of services (respectively skills) $S = \{1, ..., 6\}$ are taken into consideration. Caregivers are divided into two groups with different qualifications. Each caregiver of the first group is skilled to provide at most three

Table 6 Tuning parameters

Algorithm	Parameter	Value
GA based heuristic (Deterministic model)	P_c: crossover probability	0.8
	P_s: mutation probability	0.01
	P_{size} : population size	s^2
	Stopping criterion	$5s$
	Tournament size	$c+1$
GVNS based heuristic	Shaking phase	$h=c+1$
	Stopping criterion	$2s$
	Shaking phases order	Switch, inter-swap, intra-swap and shift
	Local search methods order	Shift, switch, inter-swap and intra-swap
Monte Carlo simulation	ϵ	0.05
	$MaxIterMCS$	100
	$MaxIterGap$	10
GA based heuristic (SPR model)	P_c: crossover probability	0.6
	P_s: mutation probability	0.08
	P_{size} : population size	100
	Stopping criterion	50
	Tournament size	$c+1$
Others	$MaxIterSyn$	$2c$
	β	100

services selected from the subset {1, 2, 3} of S. Accordingly, each caregiver of the second group is skilled for providing at most three services drawn from the subset {4, 5, 6}. Each patient requires single or double services, which are randomly drawn from $S = \{1, ..., 6\}$. 30% of patients are considered requesting double services (Mankowska et al. 2014), 50% of double services are supposed to be simultaneous and the others 50% are supposed to be received without synchronization. Time windows are of length 2 hours and are randomly placed within a daily planning period of 10 hours. Two sets of instances are generated single services (SS) and multiple synchronized services (MSS), each one contains 3 subsets and are summarized in the Table 7. '$Size$' is the number of instances in each subset. '$\#N$' is the number of patients. '$\#S$' is the total requested services by all patients. '$\#K$' is the number of caregivers available. Stochastic parameters (travel and service times) are randomly generated the same way in Shi et al. (2018). The travel time between each two patients follows a normal distribution $\widetilde{T}_{ij} \sim \mathcal{N}(c_{ij}, (\frac{c_{ij}}{3})^2)$ and the service time also follows a normal distribution $\widetilde{t}_{is} \sim \mathcal{N}(t_{is}, (\frac{t_{is}}{5})^2)$. c_{ij} and t_{is} are respectively the average of the traveling time between two patients i and j and the average time of a service operation s at patient i. $\frac{c_{ij}}{3}$ and $\frac{t_{is}}{5}$ are respectively the standard deviation values of the traveling time between two patients i and j and service operation time s for the patient i. α and γ are set to 1.

Table 7 Tested instances details

Set	Subset	Size	#N	#S	#K
SS	A	7	10	10	3
	B	7	25	25	5
	C	7	50	50	10
MSS	D	7	10	13	3
	E	7	25	33	5
	F	7	50	65	10

6.2 Computational results

Instances are generated as described above and solved within a time limit of 4 hours. LB is lower bound of the model given by CPLEX. Z is the objective function value and CPU is the computing time. For GVNS and GA based heuristics, each instance is running 10 times; the best, the worst and the average solutions are considered. CPU computing time relates to the aggregate time elapsed to solve each instance 10 times. GAP is calculated as 100%× (average—lower bound)/average). Optimal solutions reached are in boldface.

CPLEX could solve the subsets A, B, C, D and E optimally. For the subset F, only instances $F1$ and $F6$ could be solved optimally. For other instances $F2$, $F3$, $F4$, $F5$ and $F7$ a feasible solution is found with a gap respectively of 10.48%, 2.66%, 12.25%, 10.84% and 1.26%. The GVNS based heuristic could find the optimal solutions for the subsets A, B, D and E at least once among the ten runs of each instance, except for the instances $E2$. The optimal solution is found also for the instances $C4$, $C5$ and $C7$. The GA based heuristic could find the optimal solutions for the subsets A, B and D and for the instances $E1$, $E6$ and $E7$ at least once among the ten runs of each instance (see Tables 8 and 9).

The worst gaps found by the GVNS for the subsets A, B, C, D, E and F are respectively 0.99% (instance $A6$), 0.77% (instance $B6$), 3.53% (instance $C4$), 0.00% (subset D), 2.21% (instance $E2$) and 17.71% (instance $F5$). For the GA, are respectively 2.60% (instance $A6$), 2.82% (instance $B2$), 14.98% (instance $C4$), 1.04% (instance $D6$), 5.71% (instance $E3$) and 28.35% (instance $F5$). Both GVNS and GA based heuristics could find good quality solutions with a better efficiency for the GVNS (see Figs. 8 and 9).

Figure 10 shows that the minimum, first quartile and median found by CPLEX, GVNS average and GA average are very close to each other, which indicate that the performances of these methods in the first 50% of instances (subsets A, B and D) are similar. According to Tables 8 and 9, optimal solutions are reached by both heuristics at least once in 10 runs. For the subset E, both heuristics still could reach solutions very close to those found by CPLEX. GVNS showed a better efficiency and could reach 6 optimal solutions from 7 at least once in 10 runs. Figure 10 shows also that solutions reached by GA are very dispersed in the second 50% of instances (subsets C, E and F). Figures 8 and 9 show that GA found solutions for subsets C and F a little bit further from those reached by CPLEX and GVNS compared to solutions of subset E. Therefore, we can conclude that this dispersion is caused by objective function values of instances in subsets C and F. The solutions reached by GVNS and CPLEX have a very close distribution if we delete the outlier (the maximum).

In the SPR model, the transportation cost, the expected value of recourse and the objective function ($Z = cost + \widehat{\mathbb{E}(.)}$) are considered. Small instances (subset A and D) are solved by both GA and GVNS based heuristics. Solutions found by both heuristics are very close (see Table 10, the gap is computed as (average$_{GVNS}$- average$_{GA}$)/average$_{GVNS}$).

Table 8 Results of solving the set SS instances with deterministic parameters

	CPLEX				GVNS (10 runs)					Genetic Algorithm (10 runs)				
SS	LB	Z	Gap (%)	CPU	Best	Worst	Average	Gap (%)	CPU	Best	Worst	Average	Gap (%)	CPU
A1	521	**521**	0.00	1.73	**521**	**521**	**521.00**	0.00	< 1	**521**	522	521.10	0.02%	< 1
A2	715	**715**	0.00	1.49	**715**	**715**	**715.00**	0.00	< 1	**715**	754	718.90	0.54%	< 1
A3	508	**508**	0.00	1.43	**508**	**508**	**508.00**	0.00	< 1	**508**	526	509.80	0.35%	< 1
A4	817	**817**	0.00	1.52	**817**	**817**	**817.00**	0.00	< 1	**817**	**817**	**817.00**	0.00	< 1
A5	645	**645**	0.00	1.58	**645**	**645**	**645.00**	0.00	< 1	**645**	660	649.90	0.75%	< 1
A6	439	**439**	0.00	1.44	**439**	461	443.40	0.99%	< 1	**439**	461	450.70	2.60	< 1
A7	539	**539**	0.00	153.00	**539**	**539**	**539.00**	0.00	< 1	**539**	**539**	**539.00**	0.00	< 1
B1	1165	**1165**	0.00	2.66	**1165**	1178	1167.20	0.19%	5.89	**1165**	1179	1169.60	0.39%	14.96
B2	993	**993**	0.00	3.58	**993**	1003	996.60	0.36%	7.30	**993**	1070	1021.80	2.82%	14.39
B3	1131	**1131**	0.00	2.36	**1131**	**1131**	**1131.00**	0.00	10.22	**1131**	1142	1136.40	0.48%	16.05
B4	928	**928**	0.00	2.07	**928**	**928**	**928.00**	0.00	4.99	**928**	964	947.40	2.05%	15.34
B5	1064	**1064**	0.00	2.04	**1064**	**1064**	**1064.00**	0.00	5.68	**1064**	1070	1064.40	0.04%	15.69
B6	1196	**1196**	0.00	2.99	**1196**	1221	1205.30	0.77%	6.36	**1196**	1243	1215.90	1.64%	17.97
B7	1099	**1099**	0.00	2.29	**1099**	**1099**	**1099.00**	0.00	4.39	**1099**	1102	1099.60	0.05%	17.72
C1	1491	**1491**	0.00	40.70	1502	1559	1530.70	2.59%	196.59	1570	1745	1635.00	8.81%	251.65
C2	1673	**1673**	0.00	206.39	1679	1777	1728.60	3.22%	197.97	1751	1967	1901.20	12.00	254.08
C3	1567	**1567**	0.00	673.93	1570	1627	1601.90	2.18%	167.96	1650	1796	1735.40	9.70	275.72
C4	1561	**1561**	0.00	27.19	**1561**	1702	1618.10	3.53%	193.99	1621	1856	1836.00	14.98%	291.48
C5	1671	**1671**	0.00	99.44	**1671**	1784	1710.60	2.31%	234.03	1817	2047	1901.30	12.11%	338.54
C6	1658	**1658**	0.00	23.85	1683	1739	1713.40	3.23%	179.61	1738	1904	1837.40	9.76%	322.35
C7	1492	**1492**	0.00	32.76	**1492**	1595	1538.20	3.00	196.39	1601	1763	1646.80	9.40	320.45

Table 9 Results of solving the set MSS instances with deterministic parameters

MSS	CPLEX				GVNS (10 runs)					Genetic Algorithm (10 runs)				
	LB	Z	Gap (%)	CPU	Best	Worst	Average	Gap (%)	CPU	Best	Worst	Average	Gap (%)	CPU
D1	769	**769**	0.00	1.64	**769**	**769**	**769.00**	0.00	< 1	**769**	**769**	**769.00**	0.00	1.44
D2	872	**872**	0.00	1.48	**872**	**872**	**872.00**	0.00	< 1	**872**	**872**	**872.00**	0.00	1.22
D3	709	**709**	0.00	1.78	**709**	**709**	**709.00**	0.00	< 1	**709**	717	709.80	0.11	1.45
D4	938	**938**	0.00	1.56	**938**	**938**	**938.00**	0.00	< 1	**938**	**938**	**938.00**	0.00	1.25
D5	777	**777**	0.00	1.55	**777**	**777**	**777.00**	0.00	< 1	**777**	**777**	**777.00**	0.00	1.45
D6	588	**588**	0.00	1.47	**588**	**588**	**588.00**	0.00	< 1	**588**	606	594.20	1.04	1.39
D7	609	**609**	0.00	1.48	**609**	**609**	**609.00**	0.00	< 1	**609**	**609**	**609.00**	0.00	1.41
E1	1317	**1317**	0.00	18.52	**1317**	1362	1336.10	1.43	59.29	**1317**	1464	1382.70	4.75	60.88
E2	1361	**1361**	0.00	8.07	1374	1401	1391.70	2.21	57.50	1387	1447	1407.80	3.32	58.41
E3	1338	**1338**	0.00	7.15	**1338**	1383	1357.50	1.44	56.04	1353	1503	1419.10	5.71	63.79
E4	1150	**1150**	0.00	5.57	**1150**	1219	1159.40	0.81	46.39	1171	1260	1207.20	4.74	55.50
E5	1246	**1246**	0.00	3.59	**1246**	1294	1268.60	1.78	47.62	1247	1356	1286.90	3.18	64.12
E6	1251	**1251**	0.00	11.25	**1251**	1289	1259.30	0.66	51.57	**1251**	1309	1286.70	2.77	67.60
E7	1145	**1145**	0.00	5.14	**1145**	1157	1147.60	0.23	32.59	**1145**	1210	1155.10	0.87	46.88
F1	1754	**1754**	0.00	1971	1808	2002	1903.00	7.83	2766	2069	2251	2169.40	19.15	1138
F2	1668.73	1864	10.48	14400	1832	2120	1933.60	13.70	1863	1989	2312	2166.20	22.97	1197
F3	1684.98	1731	2.66	14400	1813	1839	1822.70	7.56	2052	1940	2141	2033.50	17.14	1291
F4	1721.72	1962	12.25	14400	1908	2016	1955.30	11.95	2312	2105	2269	2193.50	21.51	969
F5	1793.01	2011	10.84	14400	2040	2367	2179.00	17.71	2735	2296	2837	2502.50	28.35	1396
F6	1808	**1808**	0.00	2207	1818	1907	1853.10	2.43	2096	2035	2309	2189.00	17.41	1487
F7	1708.12	1730	1.26	14400	1738	1832	1783.10	4.21	1960	2012	2213	2098.80	18.61	1136

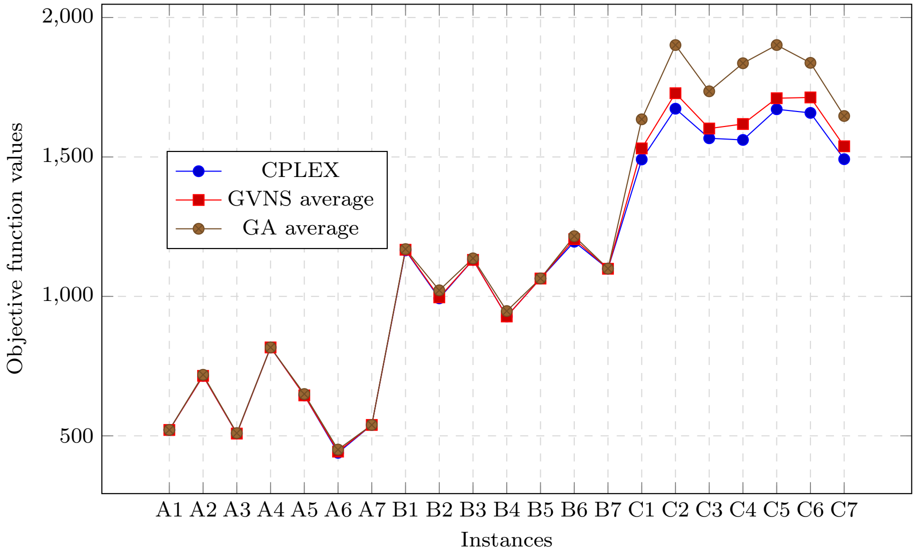


Fig. 8 Comparison of solutions found by CPLEX, GVNS and GA for the instances of the set SS with deterministic parameters

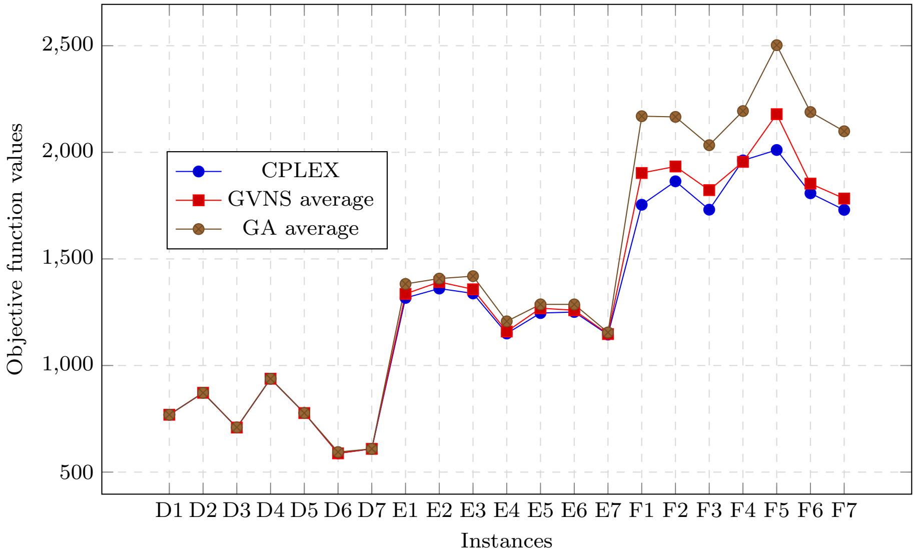


Fig. 9 Comparison of solutions found by CPLEX, GVNS and GA for the instances of the set MSS with deterministic parameters

Table 10 shows clearly that GVNS is not suitable to be combined with the simulation to solve the SPR model. The worst CPU running time jumped from 494.64 seconds (instance $A5$ of the set SS) to 1221.90 seconds (instance $D6$ of the set MSS) while the worst CPU running time of GA jumped from 134.184 seconds (instance $A4$ of the set SS) to 275.68 seconds (instance $D7$ of the set MSS). This is due to the complexity of neighborhoods sizes, which

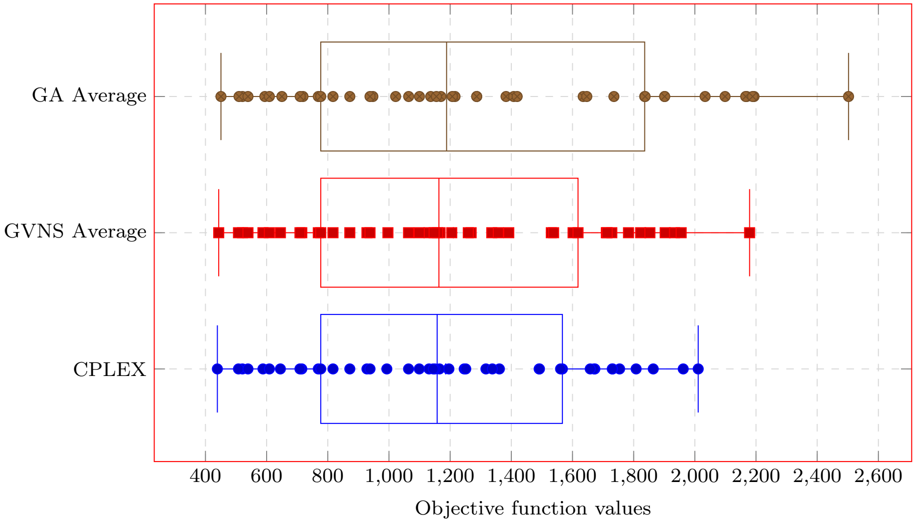


Fig. 10 Comparison of solutions found by CPLEX, GVNS and GA with deterministic parameters

Table 10 Results of solving small instances (subsets A and D) with stochastic parameters using GA and GVNS based heuristics

	GVNS (10 runs)				GA (10 runs)				
	Best	Worst	Average	CPU	Best	Worst	Average	CPU	Gap
A1	521.38	522.79	521.93	266.53	520.70	522.25	521.52	95.83	0.08%
A2	722.41	725.51	724.29	365.41	720.96	723.90	722.36	119.08	0.27%
A3	495.49	497.42	496.45	324.63	494.32	498.66	495.43	87.28	0.20%
A4	815.68	819.73	817.67	421.12	812.79	817.90	814.49	134.84	0.39%
A5	616.88	634.61	619.85	494.64	613.76	631.85	624.46	113.63	-0.74%
A6	430.95	434.63	432.94	477.53	428.35	452.80	432.83	127.54	0.03%
A7	537.15	538.80	537.96	408.06	534.82	536.95	535.92	122.84	0.38%
D1	778.68	783.76	780.38	1139.46	776.41	818.87	786.06	255.71	-0.73%
D2	872.23	872.98	872.71	621.22	872.06	872.99	872.52	184.24	0.02%
D3	688.40	698.00	693.05	918.01	683.96	697.95	694.09	192.73	-0.15%
D4	883.85	890.67	888.17	944.82	881.55	884.81	883.92	253.85	0.48%
D5	753.77	759.03	755.88	978.60	749.68	768.82	752.90	223.57	0.39%
D6	578.98	583.96	581.76	1221.90	576.52	588.86	580.31	273.40	0.25%
D7	609.43	610.66	609.94	1162.30	609.27	614.73	611.04	275.68	-0.18%

depends on the size s of the problem (number of total requested services by all patients). Three from the four proposed neighborhood structures of GVNS have a complexity of $O(s^2)$. Exploring neighborhood solutions imply running the simulation for each new generated solution to estimate the expected value of recourse, which significantly increases the CPU running time. To illustrate that, we solved the first instance of each subset ($A1$, $B1$, $C1$, $D1$, $E1$ and $F1$) by GVNS for a single iteration. Figure 11 clearly shows that the CPU running time increases monotonically with the total number of services requested by all patients. The

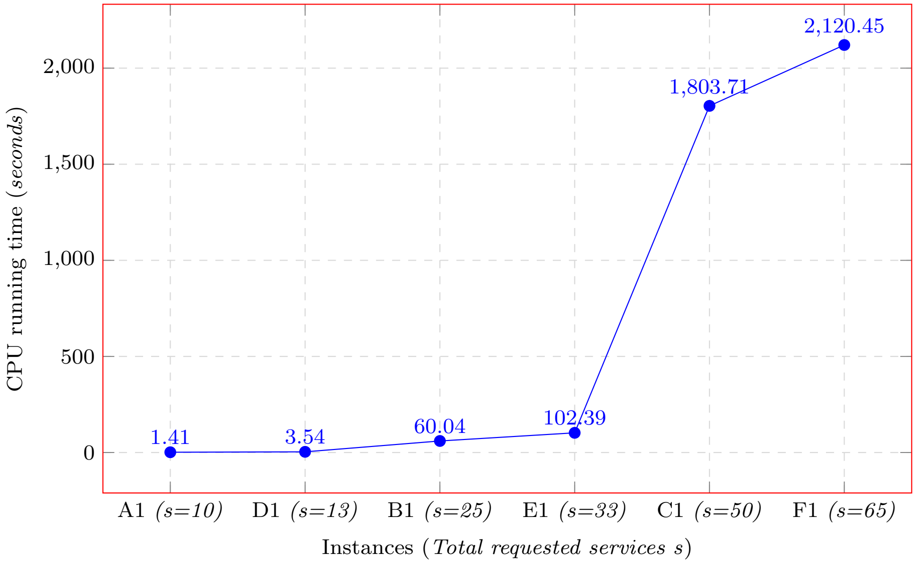


Fig. 11 CPU running times elapsed to solve the first instance of each subset using the simulation embedded into GVNS for a single iteration

CPU running time jumped from 1.41 seconds (instance $A1$ with 10 services) to 2120.45 seconds (instance $F1$ with 65 services) to run GVNS for a single iteration.

In the other hand, GA parameters are independent of the problem size. To solve medium (subsets B and E) and large (subsets C and F) instances in reasonable computational time, we fixed the population size and the stopping criterion independently of the problem size (see Table 6: SPR model). The SPR model is very complex in terms of CPU running time compared to the deterministic model, due to the expected value that must be estimated for each individual and in each iteration of the algorithm (see Table 11). In other words, the expected recourse estimation procedure is executed each time the selection-crossover-mutation mechanism is applied.

In Table 12 we compared when deterministic optimal solutions of small instances (subsets A and D) are considered as solutions for the stochastic model with solutions found by the SPR model. The first stage (caregivers' routes and their assignment to patients) is the optimal solution given by CPLEX. In the second stage, we run the simulation to estimate the expected value of recourse $\widehat{\mathbb{E}(.)}$. The total cost Z includes the transportation cost found by CPLEX and the cost of the recourse computed by the simulation, the costs of the two stages are computed independently. The GVNS (respectively GA) column contains the average solution of 10 runs for solving the SPR model and the gaps are computed as *(Z – Average)/Z*. In most cases, the values of the gaps are greater than 1, which shows that deterministic optimal solutions are not privileged in the stochastic case since the recourse is not optimized simultaneously with the transportation cost. In addition, the SPR model prioritizes solutions with higher difference between latest service times (b_i and e_k) and caregivers' completion C_{ik} to minimize the expected value of recourse. The more the margins $b_i - C_{ik}$ and $e_k - C_{ik}$ are higher the more the fluctuation of caregivers' arrival times remains robust and compatible with time windows.

Table 11 Results of solving medium and large instances with stochastic parameters using GA based heuristic

Genetic Algorithm (10 runs)									
SS	Best	Worst	Average	CPU	MSS	Best	Worst	Average	CPU
B1	1134.77	1342.94	1210.03	549.03	E1	1318.66	2030.43	1677.94	957.04
B2	1002.53	1197.89	1085.04	748.76	E2	1371.74	2180.87	1673.77	862.95
B3	1135.76	1250.39	1176.94	930.63	E3	1479.07	1767.76	1604.45	881.13
B4	942.71	1081.69	1016.55	811.72	E4	1267.52	1481.07	1361.58	925.49
B5	1076.11	1308.81	1167.70	841.30	E5	1325.51	2223.86	1636.49	836.41
B6	1158.15	1289.61	1221.40	614.21	E6	1388.26	2595.32	1750.48	818.38
B7	1116.84	1224.54	1182.24	1035.93	E7	1235.28	2688.13	1640.10	853.78
C1	1692.52	2068.92	1856.73	4714.29	F1	2942.55	4547.42	3363.43	4756.74
C2	1930.80	2244.35	2117.14	5442.66	F2	2465.93	3610.59	3114.22	5739.92
C3	1775.89	1978.25	1881.40	6233.01	F3	2253.21	3138.75	2635.77	5478.67
C4	1882.94	2368.98	2129.19	4268.39	F4	2388.92	3876.00	3148.09	4959.69
C5	1950.67	2221.42	2063.72	6063.64	F5	2861.24	4806.11	3520.34	4753.81
C6	1974.69	2135.67	2041.33	4233.30	F6	2522.03	3356.50	2908.03	5183.04
C7	1712.97	2037.94	1826.58	6020.90	F7	2018.86	3162.31	2801.73	5535.88

Table 12 Deterministic solutions with stochastic parameters and solutions found by the SPR model for the small instances

	CPLEX	Simulation		GVNS (10 runs)		GA (10 runs)	
	Cost	$\widehat{\mathbb{E}(.)}$	Z	Average	Gap	Average	Gap
A1	521	5.67	526.67	521.93	0.90%	521.52	0.98%
A2	715	20.02	735.02	724.29	1.46%	722.36	1.72%
A3	508	4.69	512.69	496.45	3.17%	495.43	3.37%
A4	817	19.8	836.8	817.67	2.29%	814.49	2.67%
A5	645	10.52	655.52	619.85	5.44%	624.46	4.74%
A6	439	1.53	440.53	432.94	1.72%	432.83	1.75%
A7	539	5.13	544.13	537.96	1.13%	535.92	1.51%
D1	769	19.65	788.65	780.38	1.05%	786.06	0.33%
D2	872	2.27	874.27	872.71	0.18%	872.52	0.20%
D3	709	40.1	749.1	693.05	7.48%	694.09	7.34%
D4	938	19.63	957.63	888.17	7.25%	883.92	7.70%
D5	777	28.38	805.38	755.88	6.15%	752.90	6.52%
D6	588	1.68	589.68	581.76	1.34%	580.31	1.59%
D7	609	5.86	614.86	609.94	0.80%	611.04	0.62%

To sum up, even though the exact methods reach the optimal solution, their computational time increases monotonically with the problem size. The two proposed heuristics GVNS and GA are able to solve large instances in a short computation time. GVNS showed the better efficiency in solving the instances for the deterministic model compared to GA. Despite this, GVNS was not suitable to be combined with simulation to solve the SPR model because three from the proposed neighborhood structures have a running time complexity of $O(s^2)$.

To overcome this limitation, we embedded the simulation into GA to solve the SPR model since its parameters do not depend on the problem size. Although the SPR model is very complex with a high run time, this model provides a robust scheduling in which uncertainties in term of traveling and caring times are taken into account. The computed expected value gives the average recourse that will be incurred for a given schedule without needing to know which patients have received the services with a tardiness and which caregivers have worked overtime.

7 Conclusion

The home health care companies aim to both minimize provided services cost and maximize patients' satisfaction. In the real world, travel and service times are not always deterministic. Uncertainties may arise and affect the overall planning and service quality would be quite poor, which will cause patients' dissatisfaction. In addition, patients need several care activities per day and some of them require to be simultaneous. In this paper, the home health care routing and scheduling problem with stochastic travel and service times as well as multiple synchronized services is addressed. The objective is to define a daily planning in which uncertainties in terms of traveling and providing services that may occur are taken into account. To that end, a two-stage stochastic programming model with recourse is proposed to minimize the transportation cost and the expected value of recourse caused by patients' delayed services and caregivers' extra working time. The first stage is to define caregivers' routes and their assignment to patients in order to compute the transportation cost. The second stage is to introduce the recourse if a solution goes against constraints related to patients' time windows and caregivers' duty length. The deterministic model is solved using CPLEX, the genetic algorithm (GA) and the general variable neighborhood search (GVNS) based heuristics. Monte Carlo simulation is used to estimate the expected value of recourse, which is embedded into the GA based heuristic to solve the stochastic model. GVNS and GA are successfully tested using several instances randomly generated from the literature. The tests prove the high performance of these two heuristics to deal with large instances in a little amount of time. GVNS and GA are able to reach optimal solutions for some instances and yield near-optimal solutions for others. The complexity of the SPR model in terms of CPU running times is significant due to the expected value that has to be estimated for each solution. Future works could be addressed to propose a robust optimization approach to deal with the HHCRSP with stochastic service and travel times and synchronized services since this approach does not require knowing the distribution of stochastic parameters. Furthermore, it would be interesting to consider multiple time windows for patients, which will be useful to minimize tardiness of services operations consequently minimize the expected value of recourse. Dynamic programming would be necessary to avoid revising the overall planning in the event that a requested service is canceled or a new risky service should be provided.

Data Availability Datasets related to this article can be found at http://dx.doi.org/10.17632/cwvgxbvw4f.1, an open-source online data repository hosted at Mendeley Data.

References

Baker J. E. (1985). Adaptive selection methods for genetic algorithms. In Proceedings of an International Conference on Genetic Algorithms and their applications, pp. 101–111. Hillsdale, New Jersey

Bazirha, M., Kadrani, A., & Benmansour, R. (2019). Daily scheduling and routing of home health care with multiple availability periods of patients. In International Conference on Variable Neighborhood Search, pp. 178–193. Springer

Bazirha, M., Kadrani, A., & Benmansour, R. (2020). Scheduling optimization of the home health care problem with stochastic travel and care times. In 2020 5th International Conference on Logistics Operations Management (GOL), pp. 1–8. IEEE

Ben-Tal, A., El Ghaoui, L., & Nemirovski, A. (2009). *Robust optimization* (Vol. 28). Princeton: Princeton University Press.

Bernard, D. G. (1955). Linear programming under uncertainty. *Management Science, 1*, 97–206.

Bertsimas, D., Brown, D. B., & Caramanis, C. (2011). Theory and applications of robust optimization. *SIAM Review, 53*(3), 464–501.

Bräysy, O., & Gendreau, M. (2005). Vehicle routing problem with time windows, part i: Route construction and local search algorithms. *Transportation Science, 39*(1), 104–118.

Burke, E. K., De Causmaecker, P., Berghe, G. V., & Van Landeghem, H. (2004). The state of the art of nurse rostering. *Journal of Scheduling, 7*(6), 441–499.

Charnes, A., & Cooper, W. W. (1959). Chance-constrained programming. *Management Science, 6*(1), 73–79.

Chepuri, K., & Homem-De-Mello, T. (2005). Solving the vehicle routing problem with stochastic demands using the cross-entropy method. *Annals of Operations Research, 134*(1), 153–181.

Cissé, M., Yalçındağ, S., Kergosien, Y., Şahin, E., Lenté, C., & Matta, A. (2017). Or problems related to home health care: A review of relevant routing and scheduling problems. *Operations Research for Health Care, 13*, 1–22.

Davis, L. (1991). Handbook of genetic algorithms

Errico, F., Desaulniers, G., Gendreau, M., Rei, W., & Rousseau, L. M. (2016). A priori optimization with recourse for the vehicle routing problem with hard time windows and stochastic service times. *European Journal of Operational Research, 249*(1), 55–66.

Gendreau, M., Laporte, G., & Séguin, R. (1996). Stochastic vehicle routing. *European Journal of Operational Research, 88*(1), 3–12.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research, 13*(5), 533–549.

Goldberg, D.E., & Deb, K (1991). A comparative analysis of selection schemes used in genetic algorithms. In Foundations of genetic algorithms, vol. 1, pp. 69–93. Elsevier

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

Holland, J. H. (1992). Genetic algorithms. *Scientific American, 267*(1), 66–73.

Juan, A. A., Faulin, J., Jorba, J., Caceres, J., & Marquès, J. M. (2013). Using parallel & distributed computing for real-time solving of vehicle routing problems with stochastic demands. *Annals of Operations Research, 207*(1), 43–65.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680.

Laporte, G., Louveaux, F., & Mercure, H. (1992). The vehicle routing problem with stochastic travel times. *Transportation Science, 26*(3), 161–170.

Li, X., Tian, P., & Leung, S. C. (2010). Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm. *International Journal of Production Economics, 125*(1), 137–145.

Liu, R., Yuan, B., & Jiang, Z. (2017). Mathematical model and exact algorithm for the home care worker scheduling and routing problem with lunch break requirements. *International Journal of Production Research, 55*(2), 558–575.

Luo, Z., Qin, H., Zhang, D., & Lim, A. (2016). Adaptive large neighborhood search heuristics for the vehicle routing problem with stochastic demands and weight-related cost. *Transportation Research Part E: Logistics and Transportation Review, 85*, 69–89.

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

Marinaki, M., & Marinakis, Y. (2016). A glowworm swarm optimization algorithm for the vehicle routing problem with stochastic demands. *Expert Systems with Applications, 46*, 145–163.

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

Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research, 24*(11), 1097–1100.

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

Redjem, R., & Marcon, E. (2016). Operations management in the home care services: a heuristic for the caregivers routing problem. *Flexible Services and Manufacturing Journal, 28*(1–2), 280–303.

Rest, K. D., & Hirsch, P. (2016). Daily scheduling of home health care services using time-dependent public transport. *Flexible Services and Manufacturing Journal, 28*(3), 495–525.

Sarasola, B., Doerner, K. F., Schmid, V., & Alba, E. (2016). Variable neighborhood search for the stochastic and dynamic vehicle routing problem. *Annals of Operations Research, 236*(2), 425–461.

Shi, Y., Boudouh, T., & Grunder, O. (2019). A robust optimization for a home health care routing and scheduling problem with consideration of uncertain travel and service times. *Transportation Research Part E: Logistics and Transportation Review, 128*, 52–95.

Shi, Y., Boudouh, T., Grunder, O., & Wang, D. (2018). Modeling and solving simultaneous delivery and pick-up problem with stochastic travel and service times in home health care. *Expert Systems with Applications, 102*, 218–233.

Taillard, E. (1990). Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational research, 47*(1), 65–74.

Tarricone, R., & Tsouros, A.D (2008). Home care in Europe: the solid facts. WHO Regional Office Europe

Taş, D., Dellaert, N., van Woensel, T., & De Kok, T. (2014). The time-dependent vehicle routing problem with soft time windows and stochastic travel times. *Transportation Research Part C: Emerging Technologies, 48*, 66–83.

Taş, D., Gendreau, M., Dellaert, N., Van Woensel, T., & De Kok, A. (2014). Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach. *European Journal of Operational Research, 236*(3), 789–799.

Trautsamwieser, A., & Hirsch, P. (2011). Optimization of daily scheduling for home health care services. *Journal of Applied Operational Research, 3*(3), 124–136.

Von Neumann, J., & Ulam, S. (1951). Monte carlo method. *National Bureau of Standards Applied Mathematics Series, 12*(1951), 36.

WHO, A., & Course, L. (2017). World report on ageing and health 2015

Yuan, B., Liu, R., & Jiang, Z. (2015). A branch-and-price algorithm for the home health care scheduling and routing problem with stochastic service times and skill requirements. *International Journal of Production Research, 53*(24), 7450–7464.